MISB MOTION IMAGERY STANDARDS BOARD **STANDARD** **Video Moving Target Indicator and Track Metadata**	**MISB ST 0903.5** **27 February 2020**

1 Scope

This standard defines two Local Set structures to deliver Video Moving Target[1] Indicator (VMTI) metadata[2] and Track metadata for populating Situational Awareness products and Common Operating Pictures with VMTI and Track overlays on Motion Imagery, and for input to tracking and data fusion systems (e.g., NATO STANAG 4676 [1] ISR tracking systems).

This standard utilizes SMPTE KLV (Key Length Value) encoding for data items and other complex data structures. Supplemental Unified Modeling Language (UML) data models provide abstraction for defining future additional encoding methods, such as Extensible Markup Language (XML). An abstract data model provides a consistent, common specification regardless of the encoding used.

Both the VMTI metadata and the Track metadata support a wide gamut of systems ranging from those producing thousands of moving targets to those producing detail about a small number of targets. Appendix A – Operational Considerations [Informative] provides informative guidance on operational considerations regarding data update frequency for conserving data bandwidth.

In practice, the VMTI Local Set can be subordinate to other MISB standards such as MISB ST 0601 [2] resulting in efficiencies though leveraging parent metadata items, or used as an independent standalone metadata set. The envisioned use case for the VTrack Local Set, on the other hand, is as an independent standalone metadata set.

2 References

MISB references cited here-in reflect versions current to the publication date of a document. In the event of a MISB document correction, the corrected document will have a single letter Minor Version appended to the Major Version number per the MISB Document Development Process [3]. For example, corrections to ST 0001.2, which has a Major Version of 2, becomes ST

[1] The term “target” rather than “object” aligns to a radar operation which discriminates a “target” from its background. Doppler radar detects moving objects resulting in “moving target indications” or MTI. In this context, “target” is used here for consistency.

2 VMTI metadata is analogous to, but distinct from, NATO STANAG 4607 radar-derived Ground Moving Target Indicator (GMTI) metadata. VMTI metadata describes imagery characteristics (e.g., color, shape, features, and identity) for which STANAG 4607 GMTI has no counterpart.

0001.2a, which includes a Minor Version of "a". The MISB will not update the referring document (this document) with the Minor Version number change. When acquiring any MISB reference listed below from an NGA repository the latest version may be either a Major or Minor Version.

[1] STANAG 4676 NATO Intelligence, Surveillance and Reconnaissance Tracking Standard, 20 May 2014.
[2] MISB ST 0601.16 UAS Datalink Local Set, Oct 2019.
[3] MISB Document Development Process, Jun 2020.
[4] MISB MISP-2020.1: Motion Imagery Handbook, Oct 2019.
[5] MISB ST 0107.4 KLV Metadata in Motion Imagery, Feb 2019.
[6] MISB ST 1201.4 Floating Point to Integer Mapping, Feb 2019.
[7] MISB ST 0603.5 MISP Time System and Timestamps, Oct 2017.
[8] MISB ST 0807.24 MISB KLV Metadata Registry, Oct 2019.
[9] MISB ST 1204.3 Motion Imagery Identification System (MIIS) Core Identifier, Feb 2020.
[10] W3C Recommendation - OWL Web Ontology Language, 2009.
[11] ISO 19156:2011 Geographical Information - Observations and Measurements (O&M).
[12] ISO 19101-1:2014 Geographic information - Reference model - Part 1: Fundamentals.
[13] ISO 19109:2015 Geographic information - Rules for application schema.
[14] ISO/IEC 9834:2014 Information technology - Procedures for the operation of object identifier registration authorities - Part 8: Generation of universally unique identifiers (UUIDs) and their use in object identifiers.
[15] IETF RFC 3986 Uniform Resource Identifier (URI): Generic Syntax, Jan 2005.

3 Revision History

Revision	Date	Summary of Changes
ST 0903.5	02/27/2020	• Document version control info added to reference section • Restructured and updated • Added Algorithm LS and Ontology LS to VMTI LS • Deprecated req's -02, -06, -07, -39, -61, -64, -73, -74, -83, -97 • Added req's -98 to -115 • New UML diagrams integrated into document sections • Changed "When" to "Where" in req -24 • Changed "element" to "item" in req's -47, -51, -58 • Added definition for Periodic Volatility • Updated references • Note: Removed UL keys for all but the top-level Local Sets. The keys do not aid in applying the standard and present more maintenance and accuracy issues than they provide benefit.

4 Acronyms

BER	Basic Encoding Rules for Object Identification
EO	Electro-Optic
EON	Electro-Optic Narrow
EOW	Electro-Optic Wide
FOV	Field of View
FPA	Focal Plane Array
FPS	Frames per second
GML	Geography Markup Language
HFOV	Horizontal Field of View
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
KLV	Key-Length-Value
LS	Local Set
LVMI	Large Volume Motion Imagery
MISB	Motion Imagery Standards Board
MISP	Motion Imagery Standards Profile
OGC	Open Geospatial Consortium
SMPTE	Society of Motion Picture and Television Engineers
ST	Standard
TLV	Tag, Length, Value
UAS	Unmanned Aerial / Airborne System
UML	Unified Modeling Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VFOV	Vertical Field of View
VMTI	Video Moving Target Indicator

5 Terms

Periodic Volatility describes data refreshed throughout a stream and valid for some period.

6 Introduction

Moving object detections and constructed tracks provide a rich source of intelligence for determining activities in motion. Standardized methods to describe and disseminate this type of information enables sharing and reuse during a mission, thereby enriching intelligence analysis.

This document defines a metadata standard for characterizing objects in Motion Imagery and associated indicators of motion. It specifies the constructs for reporting the motions of entities, the history of their motions, and the types of the entities reported. As such, the standard supports a stand-alone role in Motion Imagery-based analytics, as well as a cooperative, conjunctive analytics role with other sensor modalities and multi-source methodologies.

This standard defines both a VMTI Local Set (LS) and a Track Local Set. The VMTI LS offers a rich set of object detection qualities for identifying a multitude of objects with a frame of Motion

Imagery. The Track LS is better suited to the details of tracking one object through a sequence of Motion Imagery frames and aligns more closely in functionality with the STANAG 4676 ISR tracking standard. Each Local Set offers various strengths depending on its intended use case.

Designed to be either included within a parent metadata set such as MISB ST 0601, or function independently as a standalone system of metadata, each use case for the VMTI LS provides different benefits. Because a VMTI LS is feature rich, it potentially can increase the data bandwidth to deliver it along with accompanying Motion Imagery. As a subordinate to a parent ST 0601, however, the VMTI LS leverages existing metadata items within its parent to reduce bytes, thus being more efficient. As standalone independent metadata, the VMTI LS can still provide intelligence value without the overhead of the Motion Imagery.

A final note: the MISB prefers the term "Motion Imagery" rather than "Video" as Motion Imagery has strict criteria, such as providing intelligence information. The Motion Imagery Handbook [4] discusses these differences. As this standard preceded this preference in terminology, and to provide continuity, this document continues with video in deference to Motion Imagery.

7 KLV Data Encoding and Types

For KLV encoding of metadata, MISB ST 0107 [5] provides a set of baseline requirements.

Requirement(s)	
ST 0903.4-01	All metadata shall be expressed in accordance with MISB ST 0107.
ST 0903.5-98	Within an instance of a Local Set, tags within that Local Set shall be used only once.

The Motion Imagery Handbook details the syntax, semantics, and use of KLV items and other complex types.[3] MISB ST 0903 utilizes some of these complex types.

7.1 Simple Types

7.1.1 Floating Point Values

Data items which require a floating-point value are instead mapped to integer values according to the algorithms in MISB ST 1201 [6], which provides algorithms to convert IEEE 754 floating-point values (all precisions, i.e., 16, 32, 64 and 128 bit), including the IEEE special values of infinity and NaN, to an unsigned integer representation. The mapping is specified using the notation IMAPB(*min*, *max*, *len*), where *min* is the smallest floating point value in the range, *max* is the largest floating-point value in the range, and *len* is the number of bytes used for the unsigned integer representation. For example, IMAPB(-200, 3000, 3) specifies that a floating-point value in the range from -200 to 3000, inclusive, maps to a 3-byte integer value.

Requirement	
ST 0903.4-03	Floating point to integer mappings shall comply with MISB ST 1201.

[3] Common complex types include Local Sets, Variable-Length Packs, Defined-Length Packs, Truncation Packs, and Floating Length Packs.

7.1.2 Variable Length Values

Variable length value encoding reduces the number of bytes for unsigned integers. The specification of metadata values provides enough bytes to express their maximum value. Because leading zeroes are not significant, smaller values consume fewer bytes. For example, a value 255 represented as a four-byte value (i.e., 0x000000FF) instead uses the single byte value 0xFF.

The notation to indicate variable-length encoding is "V*max*", where *max* is the maximum number of bytes allocated. For example, "V4" indicates a variable-length encoded value specified with a minimum of one byte to a maximum of four bytes.

As an example, consider a metadata value specified as V6, that is, a maximum of 6 bytes. Coding a value = 200 requires only one byte, whereas coding a value of 123456 requires 3 bytes. Thus, there is no need to use all six bytes although specified.

Requirement(s)	
ST 0903.4-04	The number of bytes used to encode a variable length unsigned integer value shall be less than or equal to the specified maximum length.
ST 0903.4-05	The number of bytes used to encode the value zero for a variable length unsigned integer value shall be one (1).

7.1.3 Flexible Length Values

Flexible length values encode the length of its value into the value itself. BER-OID values are flexible length, because the value includes a continuation bit in each byte (see Figure 1).

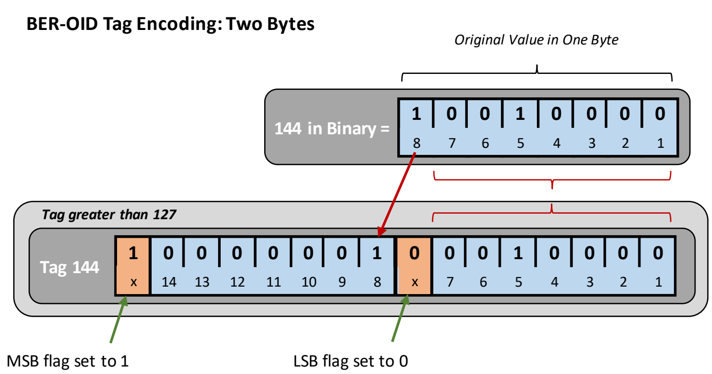

Figure 1: Example of BER-OID Encoding

7.2 Array Type

This document defines an Array type as a collection of simple data types, such as integers or floating point. Encoded as a variable-length pack, each value in an Array is of the same data type but may have a different length.

7.3 Structured Types

KLV structured types used in this standard include SMPTE Local Set (LS), Variable Length Pack (VLP), and Defined-Length Pack (DLP). See Motion Imagery Handbook for details on these types. Briefly, a LS is a collection of one or more Tag-Length-Value (TLV) items, preceded by a key (or tag) and total collection length which is the sum of bytes for all the TLV triplets. A VLP is a collection of one or more Length-Value items, preceded by a key (or tag) and total collection length which is the sum of bytes for all the Length-Value pairs in the pack. A DLP is a collection of Value items, preceded by a key (or tag) and total length in bytes of all Value items in the pack. In VLP and DLP's, items need to appear in a predefined order.

7.4 Series Type

This document defines a Series type as a collection of structured data elements, all the same type encoded as a Variable-Length Pack. Since all elements are of the same type, the element order is unimportant for decoding.

Figure 2 shows the structure of a Series type. In a Series, the "Value" is a collection of Length-Element pairs with the Length (or "L") specifying the number of bytes of the Element. The Element type is known a priori from the Key/Tag definition (in this document) enabling parsing of the Element. The number of Elements in the collection is determined by parsing the entire list in the "Value".

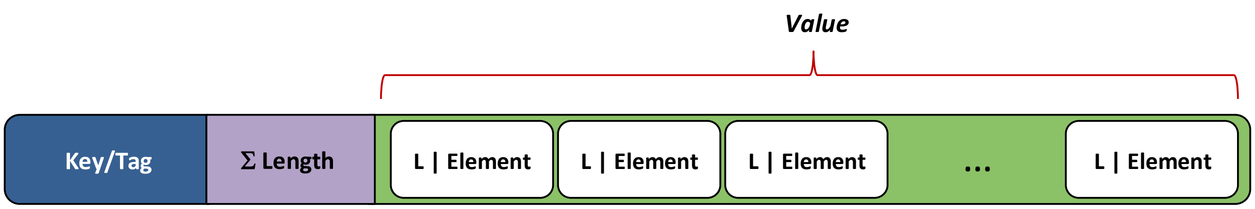

Figure 2: Series Type

In some Series the element value includes an identifier which other parts of the VMTI / VTrack Local Sets refer. These identifiers may be in the form of a BER-OID value as the first bytes of the element (i.e., the Element is a Pack) or the identifier may be the value of a Tag-Length-Value item in the element (i.e., the Element is a Local Set).

7.5 Timestamps

Timestamps across various structures in this standard utilize a Precision Time Stamp, albeit called by different names, to correlate metadata with Motion Imagery frames. MISB ST 0603 [7] defines the Precision Time Stamp as an eight-byte unsigned integer counter of the number of SI Seconds (in microseconds) which have elapsed since midnight (00:00:00), January 1, 1970(1970-01-01T00:00:00Z). Figure 3 shows an example of a VMTI LS packet containing a Precision Time Stamp.

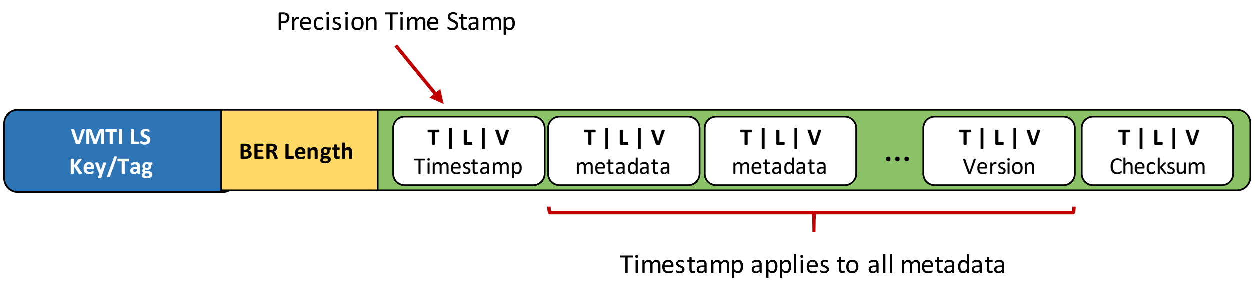

Figure 3: Packet Timestamp Example

If the VMTI LS is subordinate to a MISB ST 0601 LS (i.e., using Tag 74), a Precision Time Stamp is already present in the ST 0601 LS. In this case, a Precision Time Stamp in the VMTI LS is optional, although still recommended. If a Precision Time Stamp is available, insert it at the beginning of the value portion of the VMTI LS. While any combination of metadata items can be present in a VMTI Local Set, and most items arranged in any order, the Precision Time Stamp, if present, is the first item.

Requirement	
ST 0903.4-13	Timestamps specifying a Precision Time Stamp shall comply with MISB ST 0603.

7.6 Checksum – Error Detection

A 16-bit Checksum helps prevent the processing of erroneous metadata. The Checksum applies to a standalone VMTI LS only; when embedded into other sets, such as MISB ST 0601, the Checksum provided by the 0601 set suffices. The Checksum applies in the VTrack LS as well. The Checksum is a running 16-bit summation through the entire LS packet, starting with the 16-byte Local Set Key and ending with the Length field of the Checksum data item. The Checksum, when provided is always the *last* element in the LS. Figure 4 shows an example with the data range to which the Checksum applies. See ST 0601 for the algorithm to compute the checksum. This example also shows a VMTI LS Version Number – Tag 4 to identify the version used.

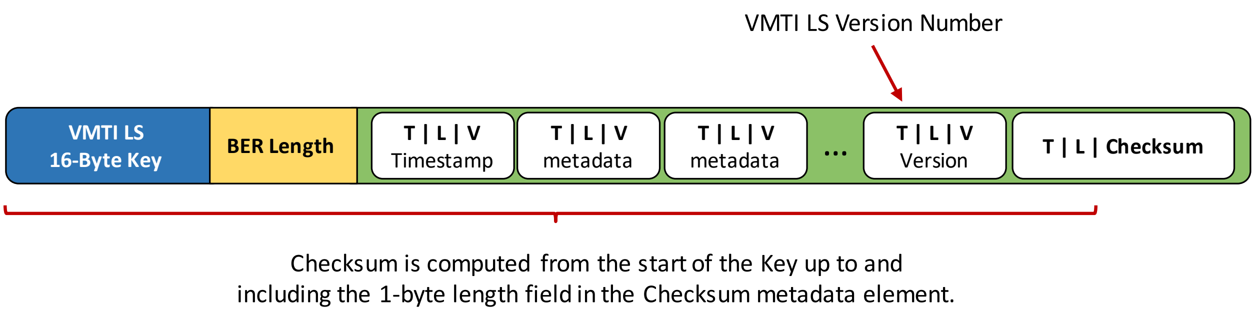

Figure 4: Example Checksum Computation Range for a Standalone VMTI LS

Requirement	
ST 0903.4-16	The Checksum shall be a 16-bit sum of all bytes in the Local Set, starting with the first byte of the 16-byte Local Set Key, up to and including the last byte of the Length field of the Checksum itself.

8 VMTI Local Set Structure

The VMTI LS contains the core information applicable to all reported phenomena within a Motion Imagery frame. Figure 5 shows a UML model of the VMTI data construct. Nomenclature and notes regarding the VMTI model:

- Orange color denotes Local Sets; Blue indicate Pack structures; green is a mix of pack/LS
- The class name reflects the collective meaning of the attributes within the class. Class attributes are the informational items defining the class content. The attribute trait Name is a single word or phrase starting with a lowercase letter and in lower camel case. The attribute trait Type defines the data representation of the attribute value once the class has been instantiated.
    - - <attribute> is for pack items e.g., -latitude in the Location Pack
    - - ##_<attribute> lists Local Set items e.g., -02_ontology in the Ontology LS
    - Multiplicities denote “Series” of Local Sets or Packs
    - Arrays (“[ ]”) denote Series of simple types such as integers

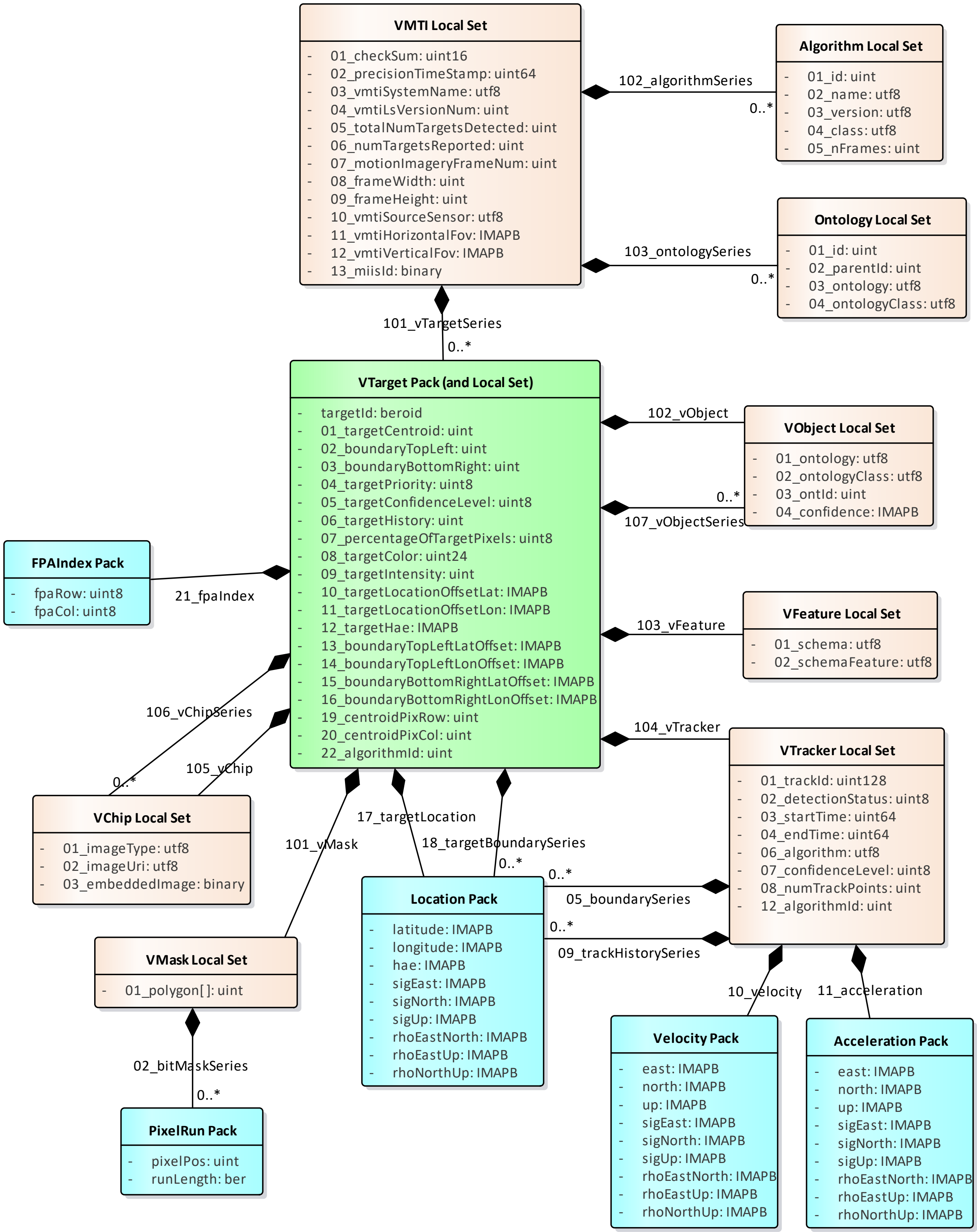

Figure 5: VMTI Local Set UML Model

Table 1 shows a high-level view of the VMTI LS. The VMTI LS supports information regarding identification and classification of multiple detections within a Motion Imagery frame.

Table 1: VMTI Local Set Structure

VMTI Local Set			Description
TLV items			TLV triplets of data items
VTargetSeries			Series of VTarget Packs
VTarget Pack$_1$	• • •	VTarget Pack$_N$	Data for one target$_1$ to target$_N$
FPA Index	• • •	FPA Index	Focal Plane Array detection occurs
VMask LS	• • •	VMask LS	Delineate target perimeter
VObject LS	• • •	VObject LS	Class or Type of target
VFeature LS	• • •	VFeature LS	Features of target
VTracker LS	• • •	VTracker LS	Track information about target
VChip LS	• • •	VChip LS	Image pixels for the target
VChipSeries[4]	• • •	VChipSeries	Series of one or more VChip LS
AlgorithmSeries			Series of detection algorithms
Algorithm LS			Algorithm description
OntologySeries			Series of ontologies used
Ontology LS			Ontology description

When used as subordinate within MISB ST 0601 (i.e., ST 0601 Tag 74), the VMTI LS expresses the frame center geographic coordinate as offsets from the frame center geographic coordinate provided in ST 0601. However, when used as a standalone independent Local Set (i.e., not subordinate to ST 0601), the VMTI LS expresses these items in absolute geographic coordinates.

Figure 6 shows the VMTI LS structure referenced subordinate to MISB ST 0601 as Tag 74. The BER Length is the sum of all data within the Value field. The Value field is composed of TLV items from the VMTI LS, and in this example, includes a VTargetSeries VLP (Tag 101 of the VMTI LS).

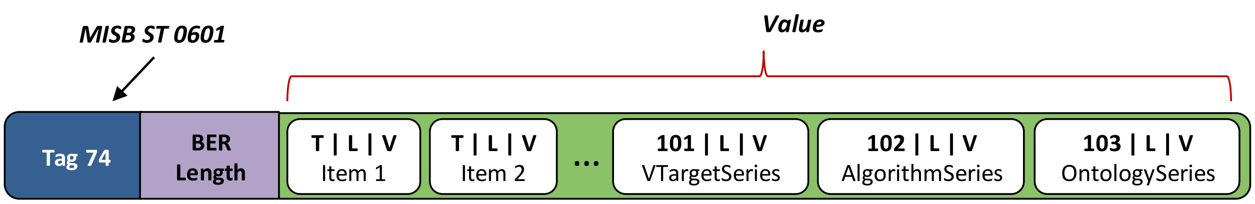

Figure 6: Data within the VMTI Structure

Figure 7 illustrates an example VMTI LS which includes several TLV items, a VTargetSeries Pack with two VTarget Packs, an AlgorithmSeries and an OntologySeries. The packet begins with a Tag 74 signaling the presence of a VMTI LS (in blue) subordinate within MISB ST 0601.

---

[4] Although permissible, it would be unusual to include both a VChip LS and a VChipSeries in a single VTarget Pack.

Following Tag 74 is a Length value, which is the sum of the lengths of all data in the VMTI LS.

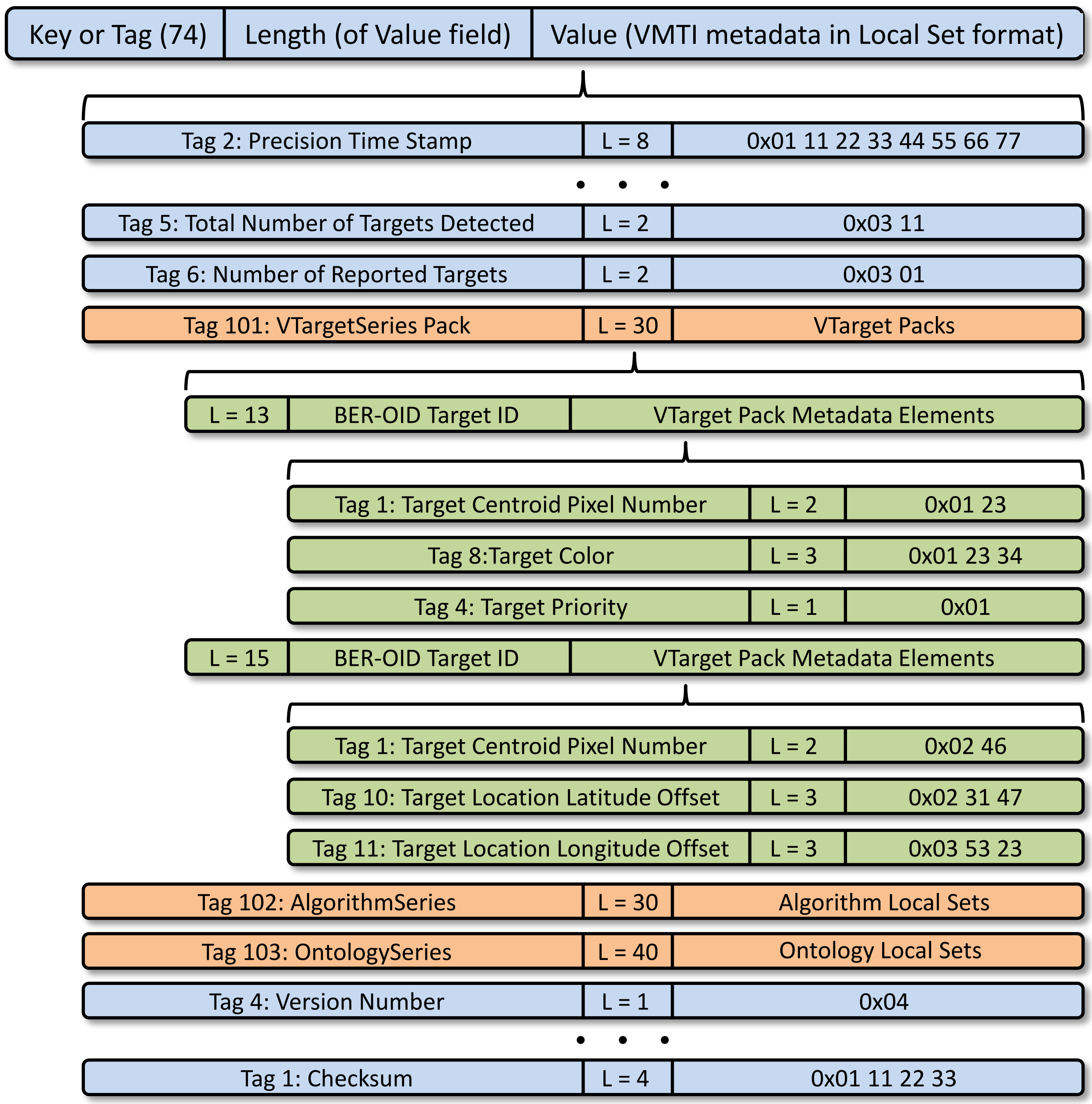

Figure 7: VMTI Local Set Example

In this example, the VMTI LS contains (among others) Tag 5 (which specifies the Total Number of Targets Detected in the Motion Imagery frame), and Tag 101 indicating a VTargetSeries Pack (orange color). The VTargetSeries Pack contains two VTarget Packs (indicated in green). The Value portion of the first VTarget Pack includes a BER-OID encoded Target ID Number, a Target Centroid Pixel Number element (Tag 1, Length = 2), a Target Color (Tag 8, Length = 3), and a Target Priority (Tag 4, Length = 1).

The Value portion of the second VTarget Pack is similar, except in addition to a Target Centroid Pixel Number it contains Target Location Latitude Offset and Target Location Longitude Offset elements (Tags 10 and 11). (For purposes of illustration, assume the Target ID Numbers each require one byte.)

Following the VTargetSeries is the AlgorithmSeries, OntologySeries, Version Number and Checksum items.

The details for the VMTI LS, VTargetSeries, and VTarget Pack follow below.

8.1 VMTI Local Set

The VMTI Local Set 16-Byte Universal Label "Key" registered in MISB ST 0807 [8] is:

06.0E.2B.34.02.0B.01.01.0E.01.03.03.06.00.00.00 (CRC 51307)

Table 2 summarizes the VMTI Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VMTI Local Set
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g., µ is micro-seconds, ° is degrees)
- "Format" indicates the item's KLV format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 2: VMTI Local Set

VMTI Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	checkSum	None	uint	2	Detects errors within a standalone VMTI LS
2	precisionTimeStamp	µs	uint	8	Microsecond count from Epoch of 1970; See MISP Time System - MISB ST 0603
3	vmtiSystemName	N/A	utf8	V32	Name and/or description of the VMTI system
4	vmtiLsVersionNum	N/A	uint	V2	Version number of the VMTI LS used to generate the VMTI metadata
5	totalNumTargetsDetected	N/A	uint	V3	Total number of targets detected in a frame. 0 represents no targets detected
6	numTargetsReported	N/A	uint	V3	Number of targets reported following a culling process
7	motionImageryFrameNum	N/A	uint	V3	Frame number identifying detected targets. Use Precision Time Stamp when available.
8	frameWidth	N/A	uint	V3	Width of the Motion Imagery frame in pixels

VMTI Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
9	frameHeight	N/A	unit	V3	Height of the Motion Imagery frame in pixels
10	vmtiSourceSensor	N/A	utf8	V128	String of VMTI source sensor. E.g. 'EO Nose', 'EO Zoom (DLTV)'
11	vmtiHorizontalFov	°	IMAPB	2	Horizontal field of view of imaging sensor input to VMTI process
12	vmtiVerticalFov	°	IMAPB	2	Vertical field of view of imaging sensor input to VMTI process
13	miisId	N/A	binary	V	A Motion Imagery Identification System (MIIS) Core Identifier conformant with MISB ST 1204
101	vTargetSeries	N/A	Series	V	VTarget Packs ordered as a Series
102	algorithmSeries	N/A	Series	V	Series of one or more Algorithm LS
103	ontologySeries	N/A	Series	V	Series of one or more Ontology LS

8.1.1 VMTI LS: Tag 1 – checkSum

The Checksum aids detection of errors within the VMTI Local Set when the LS is standalone (i.e., not as a subordinate LS to a parent metadata set). Refer to the MISB ST 0601 for the Checksum algorithm. Performed over the entire LS, the Checksum includes the 16-byte UL key and 1-byte checksum length. Value is the lower 16-bits of summation.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
0	0x01	0x02	0x0000

Requirement(s)	
ST 0903.4-15	When the VMTI LS is not embedded within a MISB ST 0601 LS (i.e., standalone), the VMTI LS shall contain a Checksum (Tag 1).
ST 0903.4-17	Where a Checksum (Tag 1) is included in a VMTI LS, it shall be the last item in the VMTI LS.

8.1.2 VMTI LS: Tag 2 – precisionTimeStamp

Defined in MISB ST 0603 the Precision Time Stamp is the number of microseconds elapsed since midnight (00:00:00), January 1,1970 not including leap seconds. If the VMTI LS is subordinate to a MISB ST 0601 LS as Tag 74, a timestamp will already be present in the MISB ST 0601 LS, and in this case, the VMTI LS Precision Time Stamp is optional – but recommended. Although not required (as some systems may not have a time source), when included a Precision Time Stamp is at the beginning of the value portion of a VMTI LS.

Some VMTI systems may not have access to a time reference; for this reason, the Precision Time Stamp is not mandatory. Data from such systems is still useful, however, even if only aligned with the Motion Imagery by time of arrival.

Note: In the absence of a Precision Time Stamp, the VMTI system should use the source (sensor) frame number to populate the Motion Imagery Frame Number metadata item – if known. When recording the frame number, it is important to account for timing differences in the sensor/compression signal path. The frame number at the sensor and the frame number after compression may be different. In most cases a frame number in a sequence of frames begins at some arbitrary number (i.e., not expected to begin at 1). If the number monotonically increases for successive frames, it will provide a metric for establishing a correspondence.

Example Value	**Example Encoded LS Value**		
April 19, 2001, 04:25:21.000000 GMT (987654321000000)	Tag	Length	Value
	0x02	0x07	0x0382 4430 F6CE 40

Requirement	
ST 0903.4-14	Where a Precision Time Stamp (Tag 2) is present in a VMTI Local Set, it shall be the first item in the VMTI Local Set.

8.1.3 VMTI LS: Tag 3 – vmtiSystemName

Name or description of the VMTI system producing the VMTI targets. A string of 32 UTF-8 one-byte characters. Note that UTF-8 allows up to four bytes per character; thus, this field can expand up to 128 bytes maximum. The field is free text.

Example Value	**Example Encoded LS Value**		
DSTO_ADSS_VMTI	Tag	Length	Value
	0x03	0x0E	0x4453 544F 5F41 4453 535F 564D 5449

8.1.4 VMTI LS: Tag 4 – vmtiLsVersionNum

Version number of the VMTI LS document used to generate the VMTI metadata. Notifies downstream clients of the LS version used to encode the VMTI metadata. Values of 1 through 65535 correspond to document revisions 1 through 65535.

Example Value	**Example Encoded LS Value**		
5 (i.e., 0903.**5**)	Tag	Length	Value
	0x04	0x01	0x05

Requirement	
ST 0903.5-99	All instances of the VMTI LS shall contain Tag 4, VMTI LS Version Number.

8.1.5 VMTI LS: Tag 5 –totalNumTargetsDetected

Total number of moving targets detected in the Frame. Particularly relevant when the number of targets reported (VMTI LS Tag 6) is less than the total number detected in the Frame. A value of zero represents no targets detected.

Example Value	**Example Encoded LS Value**

	Tag	Length	Value
28	Tag	Length	Value
	0x05	0x01	0x1C

Requirement	
ST 0903.4-18	Where the VMTI LS Total Number of Targets Detected in a Frame (Tag 5) is different from the VMTI LS Number of Reported Targets (Tag 6), the Total Number of Targets Detected in the Frame shall be specified.

8.1.6 VMTI LS: Tag 6 – numTargetsReported

Number of moving targets reported within a Frame. May be necessary (for bandwidth efficiency) to report only a subset of detected targets. Number of Reported Targets = Total Number of Targets (Tag 5) – Number of Culled Targets. The culling process is usually linked to priority value or confidence level.

Example Value	Example Encoded LS Value		
14	Tag	Length	Value
	0x06	0x01	0x0E

Requirement	
ST 0903.4-19	The VMTI LS Number of Reported Targets (Tag 6) shall always be specified.

8.1.7 VMTI LS: Tag 7 – motionImageryFrameNum

Corresponds to the Frame in which detections occur. Frame number can be used when a timestamp is not available.

Example Value	Example Encoded LS Value		
78000	Tag	Length	Value
	0x07	0x03	0x0130 B0

8.1.8 VMTI LS: Tag 8 – frameWidth

Width of the Motion Imagery frame in pixels, which corresponds to the number of pixels in a row of the image. Assumes pixels appear in row-major order. Do not use a value of zero.

An efficient method to express the location of a target within a Motion Imagery frame is with Target Centroid Pixel Number (VTarget Pack Tag 1). Computing Target Centroid Pixel Number from pixel row and column coordinates requires knowledge of the Motion Imagery frame width. Fortunately, the Motion Imagery from which VMTI information derives will always include appropriate frame size information. Converting Target Centroid Pixel Number back to row/column coordinates, however, presents an issue. Consider the case when VMTI information derived from Motion Imagery is different than that disseminated; for example, two cameras in one turret (say, narrow and wide angle) one used in the VMTI process and the other producing Motion Imagery for downstream users. A second case is VMTI information transported independent of Motion Imagery. In both instances, the Frame Width metadata specifies the frame width.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
1920	0x08	0x02	0x0780

Requirement(s)	
ST 0903.4-20	Where an associated Motion Imagery stream width is different from the Motion Imagery on which the VMTI process operates, VMTI LS Frame Width (Tag 8) shall be specified.
ST 0903.4-21	Where there is no associated Motion Imagery stream, VMTI LS Frame Width (Tag 8) shall be specified.
ST 0903.4-22	Where the VMTI LS Frame Width (Tag 8) is present, the Frame width shall be the value of the VMTI LS Frame Width.
ST 0903.4-23	Where the VMTI LS Frame Width (Tag 8) is not present, the Frame width shall be taken from the Motion Imagery Frame width.

8.1.9 VMTI LS: Tag 9 – frameHeight

Height of the Motion Imagery frame in pixels, which corresponds to the number of rows of pixels in the image. Frame Height is never a required field. Assumes pixels appear in row-major order. Do not use a value of zero.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
1080	0x09	0x02	0x0438

8.1.10 VMTI LS: Tag 10 – vmtiSourceSensor

Free text identifier of the image source sensor e.g., 'EO Nose', 'EO Zoom (DLTV)', 'EO Spotter', 'IR Mitsubishi PtSi Model 500', 'IR InSb Amber Model TBT', 'LYNX SAR Imagery', 'TESAR Imagery', etc. Identifies the VMTI processor source for systems with multiple bore-sighted sensors. Any change to the source sensor requires updating this metadata item. A string of 128 UTF-8 one-byte characters. Note that UTF-8 allows up to four bytes per character; thus, this field can expand up to 512 bytes maximum. The field is free text.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
EO Nose	0x0A	0x07	0x454F 204E 6F73 65

Requirement(s)	
ST 0903.4-24	Where the VMTI LS Source Sensor (Tag 10) is specified, it shall appear from time to time in the KLV stream as an item with Periodic Volatility.
ST 0903.4-25	Where the VMTI LS Source Sensor (Tag 10) is specified, it shall be updated at the first opportunity following a detected change.

8.1.11 VMTI LS: Tag 11 – vmtiHorizontalFov

Horizontal field of view of imaging sensor input to the VMTI process. Required only if VMTI process operates on an imaging sensor *different* than that described by the parent MISB ST 0601 LS; otherwise, the ST 0601 LS provides its Tag 16 – HFOV value. Can use HFOV (Tag 16) from ST 0601 to scale VMTI column, row coordinates.

Valid Values: The set of real numbers from 0 to 180 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
12.5 degrees	0x0B	0x02	IMAPB(0, 180, 2, 12.5) = 0x0640

Requirement	
ST 0903.4-26	Where the Motion Imagery used in the VMTI process is different from the Motion Imagery produced for downstream users, the VMTI Sensor Horizontal Field of View (Tag 11) shall be populated.

VMTI Sensor Horizontal Field of View Tag 11 (HFOV) should only be populated when the Motion Imagery input to the VMTI process is *different* from that streamed with MISB ST 0601 data; otherwise, ST 0601 Tag 16 – HOV provides this value. Under these circumstances and when the two sensors share a common boresight, the ratio (k_x) of the HFOV value in the VMTI LS to the value recorded in the MISB ST 0601 can be used to scale the VMTI *Column* coordinate for the display of the streamed Motion Imagery. The scaling is:

$$x_2 = k_x\left(x_1 - \left(\frac{Frame\_Width}{2}\right)\right) + \frac{Frame\_Width}{2}$$

Where: x_1 is the original *Column* coordinate of the target extracted from the target pixel number
x_2 is the scaled *Column* coordinate of the target in the Motion Imagery
k_x is the scaling factor calculated according to the following equation:

$$k_x = \frac{\tan\left(\frac{1}{2}\theta_{H1}\right)}{\tan\left(\frac{1}{2}\theta_{H2}\right)}$$

Where: θ_{H1} is the HFOV of the original (VMTI) sensor (degrees)
θ_{H2} is the HFOV of the sensor to which the targets are those scaled (degrees)

In most cases k_x will be enough to perform scaling in the *Column* direction (*Row* scaling uses frame height rather than frame width – see Tag 12). The VFOV element in VMTI LS Tag 12 provides for those cases where the aspect ratio of the two sensors is different (for example 4:3 and 16:9).

These equations are valid if the frame width of the two sensors is the same. If not, the equations become more complex. When the streamed Motion Imagery is from a narrow field of view sensor, and the VMTI process is run on Motion Imagery from a bore-sighted wide field of view sensor, it can be expected that moving targets will be outside the frame boundaries of the streamed Motion Imagery. Under these circumstances, the display device could add a blank border around the active area and present the moving targets to the operator – albeit without the underlying Motion Imagery content. Alternatively, the display device could present highlights around the perimeter of the frame to indicate movers that are outside the frame along with their radial direction from the boresight.

8.1.12 VMTI LS: Tag 12 – vmtiVerticalFov

Vertical field of view of imaging sensor input to VMTI process. Required only if VMTI process operates on an imaging sensor *different* from that described by the parent MISB ST 0601 LS; otherwise, the ST 0601 LS provides its Tag 167– VFOV value. Can use with VFOV (Tag 17) from ST 0601 to scale VMTI column, row coordinates. Typically required only to account for aspect ratio variation.

Valid Values: The set of real numbers from 0 to 180 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.0 degrees	0x0C	0x02	IMAPB(0, 180, 2, 10.0) = 0x0500

Requirement	
ST 0903.4-27	Where the Motion Imagery used in the VMTI process is different from the Motion Imagery produced for downstream users, the VMTI Sensor Vertical Field of View (Tag 12) shall be populated.

VMTI Sensor Vertical Field of View Tag 12 should only be populated when the Motion Imagery input to the VMTI process is *different* from that streamed with the MISB ST 0601 data. Under these circumstances, and when the two sensors share a common boresight, the ratio (k_y) of the VFOV value in the VMTI LS to the value recorded in the MISB ST 0601 can be used to scale the VMTI *Row* coordinate for the display of the streamed Motion Imagery. The scaling is:

$$y_2 = k_y\left(y_1 - \left(\frac{Frame\_Height}{2}\right)\right) + \frac{Frame\_Height}{2}$$

Where: y_1 is the original *Row* coordinate of the target extracted from the target pixel number
y_2 is the scaled *Row* coordinate of the target in the Motion Imagery
k_y is the scaling factor calculated according to the following equation:

$$k_y = \frac{\tan\left(\frac{1}{2}\theta_{V1}\right)}{\tan\left(\frac{1}{2}\theta_{V2}\right)}$$

Where: θ_{V1} is the VFOV of the original (VMTI) sensor (degrees)
θ_{V2} is the VFOV of the sensor to which the targets are those scaled (degrees)

In most cases k_x (see Tag 11) will be enough to perform scaling in the *Row* direction. The VFOV element in VMTI LS Tag 12 provides for those cases where the aspect ratio of the two sensors is different (for example 4:3 and 16:9). These equations are valid if the frame height of the two sensors is the same. If not, the equations become more complex.

8.1.13 VMTI LS: Tag 13 – miisId

Provides a unique identifier for the Motion Imagery stream. Appearance of this item in the metadata stream is desirable but not mandatory. If the VMTI LS is subordinate to a MISB ST 0601 LS under Tag 74, a MIIS Core Identifier may already be present in the ST 0601 LS. In this case, omit this item in the VMTI LS.

Valid Values: A value conformant with MISB ST 1204 [9].

8.1.14 VMTI LS: Tag 101 – vTargetSeries

VTargetSeries is a Series type which contains VTarget Packs only. The Length field for the series is the sum of all the data in the VTargetSeries Value field. The Value field is comprised of one or more VTarget Packs, each of which can be of a different size (thereby including different information) parsed according to the Length provided for each VTarget Pack.

Figure 8 illustrates a VTargetSeries within a VMTI LS. Here the VMTI LS's first data is a TLV item indicating Tag 5 "Total Number of Targets Detected in the Frame", a Length of 1 byte, and a Value of 2 indicating two VTarget Packs. Tag 101 indicates a VTargetSeries with a full Length shown as ΣL (i.e., sum of all data that follows). Two VTarget Packs within the VTargetSeries follow; the Length (i.e., L) in each VTarget Pack specifies the pack's length for parsing.

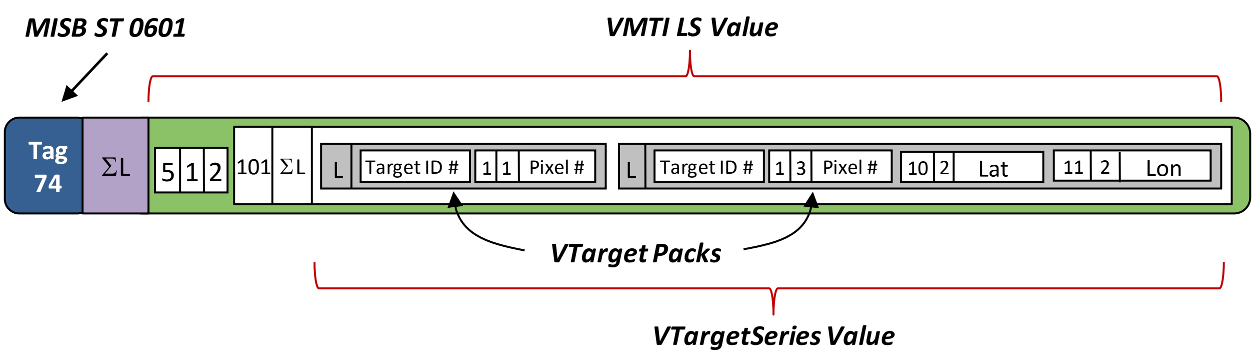

Figure 8: VMTI Local Set VTargetSeries Example

Requirement	
ST 0903.4-08	Each VTarget Pack in a VTargetSeries shall contain metadata for a single target.

Example Value	Example Encoded LS Value		
	Tag	Length	Value (Length, VTarget Pack Data)
Three VTarget Packs with Target IDs of 1, 2 and 3	0x65	0xL0	L1 0x01… L2 0x02… L3 0x03…

Where L0 is the Length of all VTarget Packs in the VTargetSeries; and L1, L2, L3 are the Lengths of each VTarget Pack. BER-OID Target IDs (i.e., 0x01, 0x02, 0x03) and the remaining VTarget Pack Data follows each length (i.e., L1, L2, L3) .

8.1.15 VMTI LS: Tag 102 – algorithmSeries

AlgorithmSeries is a Series type which contains one or more Algorithm Local Sets. Within the VMTI LS the Tag for the algorithmSeries is Tag 102. The Length field for the pack is the sum of all the data in the algorithmSeries Value field. The Value field is comprised of one or more Algorithm LS, each of which can be of a different size (thereby including different information) parsed according to the Length provided for each LS.

The algorithmSeries enables assigning an algorithm to a detected target or a generated track, whereas the algorithm item in the VTracker LS only assigns an algorithm to a track. The recommendation is to use algorithmSeries because 1) it affords specifying a multitude of algorithms as needed, and 2) definition at the parent VMTI LS level is more bandwidth efficient when an algorithm applies to more than one detection.

8.1.16 VMTI LS: Tag 103 – ontologySeries

OntologySeries is a Series type which contains one or more Ontology Local Sets. Within the VMTI LS the Tag for the ontologySeries is Tag 103. The Length field for the pack is the sum of all the data in the ontologySeries Value field. The Value field is comprised of one or more Ontology LS, each of which can be of a different size (thereby including different information) parsed according to the Length provided for each LS.

The ontologySeries enables assigning multiple ontologies to a detected target, whereas the ontology items in the VObject LS referenced by the VTarget LS only allow a single ontology per detected target. The recommendation is to use ontologySeries because 1) it affords specifying a multitude of ontologies as needed, and 2) definition at the parent VMTI LS level is more bandwidth efficient when an ontology applies to more than one target.

8.2 VTarget Pack

As discussed, the VTargetSeries contains one or more VTarget Packs. The VTarget Pack has only two data values: a BER-OID encoded Target Number and one or more TLV items as shown in Figure 9.

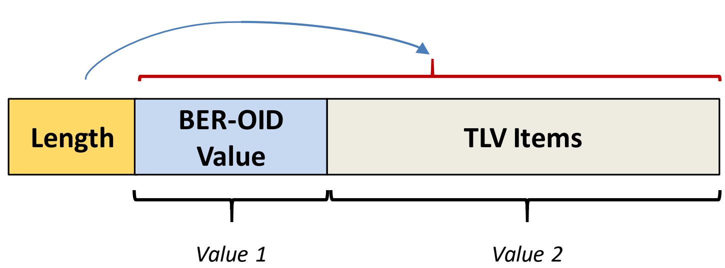

Figure 9: Structure of a VTarget Pack

The VTarget Pack has a defined order; however, Value 1 may have variable size, and Values 2 may have a different number of items per usage. Thus, the pack Length, which is a sum of all data in Value 1 and Value 2, can vary at runtime (when instantiated).

At a high level, Table 3 lists the data items, Locals Sets and Packs possible within a VTarget Pack.

Table 3: VTarget Pack Composition

	VTarget Pack Items and Complex Structures	
	Data	**Purpose**
Value 1	BER-OID value	Target ID Number
Value 2	Items	Various TLV items e.g., position, priority, color, height, etc.
	FPA Index	Specifies column/row of the Focal Plane Array (FPA) target occurs
	VMask LS	To delineate the perimeter of a target
	VObject LS	Specify the class or type of target according to defined ontology
	VFeature LS	Specific features of a target
	VTracker LS	Provides track information about a target
	VChip LS	Provides the pixel values of a target
	VChipSeries	Series of one or more VChip LS
	VObjectSeries	Series of one or more VObject LS
	Target Location	Location of target
	TargetBoundarySeries	Series of boundaries around targets

Figure 10 shows an example VTarget Pack, where the first value is the Target ID Number encoded as BER-ID. Subsequent selections available in the VTarget Pack follow. As not all VTarget Packs may have the same information, in general they will be of varying size. The Pack Length (shown in yellow) accounts for the cumulative length in bytes across all the packs.

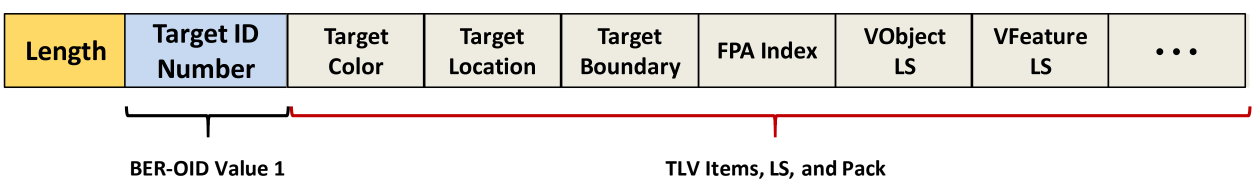

Figure 10: VTarget Pack Example

Note that the VTarget Pack is unusual in that the first value in the pack's value field is the BER-OID encoded Target ID Number, <u>without</u> a Tag or a Length, yet the remaining items adhere to the conventional TLV structure, <u>with</u> a Tag and a Length. Because the Target ID Number is mandatory, whereas other elements are optional, this construct saves bandwidth. In the simplest case, a VTarget Pack can consist of just the Target ID Number and the Target Centroid Pixel Number.

Requirement(s)	
ST 0903.4-09	The first data in the value field of a VTarget Pack shall be a BER-OID-encoded value that represents the Target ID Number of a target.
ST 0903.4-10	Where a VTarget Pack is used, at least one Tag-Length-Value item shall follow the Target ID Number.
ST 0903.4-11	All data within a VTarget Pack, excluding the Target ID Number, shall be items encoded as Tag-Length-Value items.
ST 0903.4-12	The Length field of a VTarget Pack shall be BER-encoded.

Table 4 summarizes the VTarget Pack. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VTarget Pack
- "Name" is the label associated with the Tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g., ° is degrees, m is meters)
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 4: VTarget Pack (and Local Set)

VTarget Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	targetId	N/A	uint	V3	Mandatory BER-OID encoded first value in a VTarget Pack
1	targetCentroid	N/A	uint	V6	Defines the position of the target within the Motion Imagery frame in pixels
2	boundaryTopLeft	N/A	uint	V6	Position in pixels of the top left corner of the target bounding box within the Motion Imagery Frame
3	boundaryBottomRight	N/A	uint	V6	Position in pixels of the bottom right corner of the target bounding box within the Motion Imagery Frame
4	targetPriority	N/A	uint	1	Priority or validity of target based on criteria within the VMTI system
5	targetConfidenceLevel	N/A	uint	1	Confidence level of target based on criteria within the VMTI system
6	targetHistory	N/A	uint	V2	Number of previous times the same target detected
7	percentageOfTargetPixels	N/A	uint	1	Percentage of pixels within the bounding box detected to be target pixels rather than background pixels
8	targetColor	N/A	uint	3	Dominant color of the target

VTarget Pack					
Tag ID	Name	Units	Format	Len	Description
9	targetIntensity	N/A	uint	V3	Dominant Intensity of the target
10	targetLocationOffsetLat	°	IMAPB	3	Latitude offset for target from frame center latitude (used with MISB ST 0601)
11	targetLocationOffsetLon	°	IMAPB	3	Longitude offset for target from frame center longitude (used with MISB ST 0601)
12	targetHae	m	IMAPB	2	Height of target in meters above WGS84 Ellipsoid
13	boundaryTopLeftLatOffset	°	IMAPB	3	Latitude offset for top left corner of target bounding box
14	boundaryTopLeftLonOffset	°	IMAPB	3	Longitude offset for top left corner of target bounding box
15	boundaryBottomRightLatOffset	°	IMAPB	3	Latitude offset for bottom right corner of target bounding box
16	boundaryBottomRightLonOffset	°	IMAPB	3	Longitude offset for bottom right corner of target bounding box
17	targetLocation	N/A	Location	V	Location of the target (latitude, longitude, & height above WGS84 Ellipsoid), with sigma and rho values
18	targetBoundarySeries	N/A	Series	V	Boundary around the target
19	centroidPixRow	N/A	uint	V4	Specifies the row in pixels of the target centroid within the Motion Imagery Frame
20	centroidPixCol	N/A	uint	V4	Specifies the column in pixels of the target centroid within the Motion Imagery Frame
21	fpaIndex	N/A	dlp	2	Specifies the column and the row of a sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs
22	algorithmId	N/A	uint	V3	Identifier indicating which algorithm in Algorithm Series detected this target
101	vMask	N/A	Local Set	V	Local Set to include a mask for delineating the perimeter of the target
102	vObject	N/A	Local Set	V	Local Set to specify the class or type of a target
103	vFeature	N/A	Local Set	V	Local Set to include features about the target
104	vTracker	N/A	Local Set	V	Local Set to include track information about the target

VTarget Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
105	vChip	N/A	Local Set	V	Local Set to include underlying pixel values for the target
106	vChipSeries	N/A	Series	V	Series of one or more VChip LS
107	vObjectSeries	N/A	Series	V	Series of one or more VObject LS

8.2.1 VTarget Pack: targetId Number

Target ID Number is mandatory and always comes first in a VTarget Pack. It is BER-OID encoded to convey the length but has no Tag field. BER-OID encoding allocates the upper bit of each byte to indicate "last Byte" (set =0) or continuation to next byte (set=1). Three (3) bytes provide up to 21 bits for the ID value, for a maximum value of 2,097,151. Target ID Number uniquely identifies a target across multiple frames until the New Detection Flag (Tag 6 within the VTarget Pack) resets the number. Sophisticated VMTI systems may use the same Target ID Number to identify a common target detected by different sensors, thereby correlating targets detected simultaneously by different sensor systems.

Example Value	Example Encoded Value		
	Tag	Length	Value (BER-OID)
1234	N/A	N/A	0x8952

Requirement	
ST 0903.4-28	To the extent possible a VTarget Pack Target ID Number shall uniquely identify a given target.

8.2.2 VTarget Pack: Tag 1 – targetCentroid

Specifies the position of the target centroid within a frame (see below for equations). Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom The calculation of the pixel number uses the equation: Column + ((Row-1) x Frame Width)). The top left pixel of a frame equates to (Column, Row) = (1, 1) and pixel number 1. The Frame Width is the value of VMTI LS Tag 8, if present. If it is not present, then the Frame Width comes from the underlying Motion Imagery. In the absence of underlying Motion Imagery, VMTI LS Tag 8 needs to be present. Range: All integer values from 1 to 0xFFFFFFFFFFFF (281,474,976,710,655).

Two representations of the Target Centroid Pixel Number are possible specified using either the Target Centroid Pixel Number (Tag 1), or the pair Target Centroid Pixel Row (Tag 19) and Target Centroid Pixel Column (Tag 20).

Example Value	Example Encoded LS Value		
	Tag	Length	Value
409,600	0x01	0x03	0x0640 00

Equations for calculating the coordinates of the centroid (geometric center) (x_c, y_c) of a non-intersecting closed polygon with n vertices:

$$x_c = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i),\quad y_c = \frac{1}{6A}\sum_{i=0}^{n-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1} y_i)\,\mathrm{x_c} = \frac{1}{6\mathrm{A}}\sum_{\mathrm{i=0}}^{\mathrm{n\text{-}1}}(\mathrm{x_i} + \mathrm{x_{i+1}})(\mathrm{x_i y_{i+1}\text{-}x_{i+1} y_i}),\mathrm{y_c} = \frac{1}{6\mathrm{A}}\sum_{\mathrm{i=0}}^{\mathrm{n\text{-}1}}(\mathrm{y_i} + \mathrm{y_{i+1}})(\mathrm{x_i y_{i+1}\text{-}x_{i+1} y_i})$$

Where, $$A = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)\,\mathrm{A} = \frac{1}{2}\sum_{\mathrm{i=0}}^{\mathrm{n\text{-}1}}(\mathrm{x_i y_{i+1}\text{-}x_{i+1} y_i})$$

As a practical matter, calculation of the exact centroid is probably not necessary, although it is not precluded. The centroid of a simple bounding box might be adequate.

Requirement	
ST 0903.4-29	At least one representation of VTarget Pack Target Centroid Pixel Number (Tag 1) shall be present. The Target Centroid Pixel Number can be specified using either the data element Target Centroid Pixel Number (Tag 1)or the pair of data elements Target Centroid Pixel Row (Tag 19)and Target Centroid Pixel Column (Tag 20).

8.2.3 VTarget Pack: Tag 2 – boundaryTopLeft

Specifies the position of the top left corner of the target bounding box within a frame as a pixel number. Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom. The calculation of the pixel number uses the equation: *Column* + ((*Row*-1) x Frame Width)). The top left pixel of the frame equates to (*Column*, *Row*) = (1, 1) and pixel number 1. The Frame Width comes from VMTI LS Tag 8, if present. If it is not present, then the Frame Width comes from the underlying Motion Imagery. In the absence of underlying Motion Imagery, VMTI LS Tag 8 needs to be present.

It is important for bit efficiency to rely on variable length payloads for this value.

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
409,600	0x02	0x03	0x0640 00

8.2.4 VTarget Pack: Tag 3 – boundaryBottomRight

Specifies the position of the bottom right corner of the target bounding box within the frame as a pixel number. Numbering commences with 1, at the top left pixel, and proceeds from left to right, top to bottom. The calculation of the pixel number uses the equation: *Column* + ((*Row*-1) x Frame Width)). The top left pixel of the frame equates to (*Column*, *Row*) = (1, 1) and pixel number 1. The Frame Width comes from VMTI LS Tag 8, if present. If it is not present, then the Frame Width comes from the underlying Motion Imagery. In the absence of underlying Motion Imagery, VMTI LS Tag 8 needs to be present.

It is important for bit efficiency to rely on variable length payloads for this value.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
409,600	0x03	0x03	0x0640 00

8.2.5 VTarget Pack: Tag 4 – targetPriority

Provides systems downstream a means to intelligently cull targets for a given frame.

For example, VMTI processors may generate thousands of hits.

Valid Values: 1 to 255, where 1 is the highest priority.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
27	0x04	0x01	0x1B

8.2.6 VTarget Pack: Tag 5 – targetConfidenceLevel

Confidence level, expressed as a percentage, based on criteria within the VMTI system. Target(s) with the highest confidence may not have the highest priority value. Potential for use in limited bandwidth scenarios to only send highest confidence targets. Multiple targets may have the same confidence level. Range 0 to 100, where 100 percent is the highest confidence. Although a confidence level of 0 percent indicates no confidence that a detection is a potential target. A target detected with a high confidence may be a low priority target.

Valid Values: The set of integer values from 0 to 100 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
80%	0x05	0x01	0x50

8.2.7 VTarget Pack: Tag 6 – targetHistory

Primarily indicates detection of a new target or reuse of a previous Target ID Number. Also provides the ability to indicate target persistence i.e., the number of previous times the same target is detected and may provide useful context when a target reappears after no detection for a significant time. There is no requirement that detections be in consecutive frames.

Valid Values: 0 to 65535 frames, where a value of 0 denotes the target as a new detection.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
2765	0x06	0x02	0x0ACD

8.2.8 VTarget Pack: Tag 7 – percentageOfTargetPixels

The percentage of pixels within the bounding box detected to be target pixels rather than background pixels. Range 1 to 100, where 100 signifies that the target completely fills the bounding box. Use of the VMask, VChip, VObject, or VFeature Local Sets recommended where more detail about a target is necessary. If a detection has occurred, the size of the bounding box should be such that a non-zero percentage of pixels overlaps the target.

Valid Values: The set of integer numbers from 1 to 100 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
50%	0x07	0x01	0x32

8.2.9 VTarget Pack: Tag 8 – targetColor

Dominant color of the target expressed using RGB color values. General mapping of any multispectral dataset to an RGB value. Primary use when transmitting metadata in the absence of the underlying Motion Imagery. VFeature LS provides more comprehensive color information. Represents the RGB color value with: First byte = Red, Second byte = Green, Third byte = Blue.

Valid Values: All integer values from 0 to 0xFF (255) for each of three one-byte fields.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
(Red, Green, Blue) = (218, 165, 32) (i.e., “Goldenrod”)	0x08	0x03	0xDAA5 20

8.2.10 VTarget Pack: Tag 9 – targetIntensity

Dominant intensity of the target with dynamic range up to 24 bits. For use when transmitting metadata in the absence of the underlying Motion Imagery. Primarily designed for Infrared systems that may detect targets with greater than 8-bits per pixel dynamic range and transmit the signal at a lower dynamic range. Intensity value meaning (i.e., White-Hot or Black-Hot)is consistent with IR Polarity specified in MISB ST 0601.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
13140	0x09	0x02	0x3354

8.2.11 VTarget Pack: Tag 10 – targetLocationOffsetLat

Latitude offset for target from Frame Center Latitude (MISB ST 0601), based on WGS84 ellipsoid. Has meaning only when embedding the VMTI LS in ST 0601 LS. The Target Location Latitude Offset adds to the Frame Center Latitude metadata item from the parent ST 0601 to determine the Latitude of the target. Both data items need to be in decimal representation prior to addition to determine the actual measured or calculated Motion Imagery target location.

Target Location Latitude Offset has a real earth coordinate represented by a latitude-longitude pair. Target locations that lie above the horizon do not correspond to a point on the earth. Also, target locations may lie outside of the mapped range. Both cases should either not be reported or be reported as an “error”. Conversion: IMAPB (-19.2, 19.2, 3).

Valid Values: The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.00 Degrees	0x0A	0x03	IMAPB(-19.2, 19.2, 3, 10.00) = 0x3A66 67

Requirement	
ST 0903.4-30	VTarget Pack Target Location Latitude Offset (Tag 10) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.12 VTarget Pack: Tag 11 – targetLocationOffsetLon

Longitude offset for target from Frame Center Longitude (MISB ST 0601), based on WGS84 ellipsoid. Has meaning only when embedding the VMTI LS in ST 0601 LS. The Target Location Longitude Offset adds to the Frame Center Longitude metadata item from the parent ST 0601 to determine the Longitude of the target. Both data items need to be in decimal representation prior to addition to determine the actual measured or calculated Motion Imagery target location.

Target Location Longitude Offset has a real earth coordinate represented by a latitude-longitude pair. Target locations that lie above the horizon do not correspond to a point on the earth. Also, target locations may lie outside of the mapped range. Both cases should either not be reported or be reported as an "error". Conversion: IMAPB(-19.2, 19.2, 3).

Valid Values: The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
12.00 Degrees	0x0B	0x03	IMAPB(-19.2, 19.2, 3, 12.00) = 0x3E66 67

Requirement	
ST 0903.4-31	VTarget Pack Target Location Longitude Offset (Tag 11) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.13 VTarget Pack: Tag 12 – targetHae

Height of the target expressed as height in meters above the WGS84 ellipsoid (HAE). Conversion: IMAPB(-900, 19000, 2).

Valid Values: The set of real numbers from -900 to 19,000 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10,000 meters	0x0C	0x02	IMAPB(-900, 19000, 2, 10000) = 0x2A94

8.2.14 VTarget Pack: Tag 13 – boundaryTopLeftLatOffset

Latitude offset for top left corner of target bounding box from Frame Center Latitude (MISB ST 0601), based on WGS84 ellipsoid. Use with Frame Center Latitude. Added to the Frame Center Latitude from the parent ST 0601 packet to determine the Latitude of the top left corner of the target bounding box. Convert both data items to decimal representation prior to addition to determine the actual measured or calculated Motion Imagery bounding box corner location. Bounding box corners that lie above the horizon do not correspond to points on the earth. Bounding box corners may lie outside of the mapped range. Both cases should either not be reported or be reported as an "error". Conversion: IMAPB (-19.2, 19.2, 3).

Valid Values: The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.00 Degrees	0x0D	0x03	IMAPB(-19.2, 19.2, 3, 10.00) = 0x3A66 67

Requirement	
ST 0903.4-32	VTarget Pack Bounding Box Top Left Latitude Offset (Tag 13) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.15 VTarget Pack: Tag 14 – boundaryTopLeftLonOffset

Longitude offset for top left corner of target bounding box from Frame Center Longitude (MISB ST 0601), based on WGS84 ellipsoid. Use with Frame Center Longitude. Added to the Frame Center Longitude from the parent ST 0601 packet to determine the Longitude of the top left corner of the target bounding box. Convert both data items to decimal representation prior to addition to determine the actual measured or calculated Motion Imagery bounding box corner location. Bounding box corners that lie above the horizon do not correspond to points on the earth. Bounding box corners may lie outside of the mapped range. Both cases should either not be reported or be reported as an "error". Conversion: IMAPB(-19.2, 19.2, 3).

Valid Values: The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.00 Degrees	0x0E	0x03	IMAPB(-19.2, 19.2, 3, 10.00) = 0x3A66 67

Requirement	
ST 0903.4-33	VTarget Pack Bounding Box Top Left Longitude Offset (Tag 14) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.16 VTarget Pack: Tag 15 – boundaryBottomRightLatOffset

Latitude offset for bottom right corner of target bounding box from Frame Center Latitude (MISB ST 0601), based on WGS84 ellipsoid. Use with Frame Center Latitude. Added to the Frame Center Latitude from the parent ST 0601 packet to determine the Latitude of the bottom right corner of the target bounding box. Convert both data items to decimal representation prior to addition to determine the actual measured or calculated Motion Imagery bounding box corner location. Bounding box corners that lie above the horizon do not correspond to points on the earth. Bounding box corners may lie outside of the mapped range. Both cases should either not be reported or be reported as an "error". Conversion: IMAPB(-19.2, 19.2, 3).

Valid Values: The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.00 Degrees	0x0F	0x03	IMAPB(-19.2, 19.2, 3, 10.00) = 0x3A66 67

Requirement	
ST 0903.4-34	VTarget Pack Bounding Box Bottom Right Latitude Offset (Tag 15) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.17 VTarget Pack: Tag 16 – boundaryBottomRightLonOffset

Longitude offset for bottom right corner of target bounding box from Frame Center Longitude (MISB ST 0601), based on WGS84 ellipsoid. Use with Frame Center Longitude. Added to the Frame Center Longitude from the parent ST 0601 packet to determine the Longitude of the bottom right corner of the target bounding box. Convert both data items to decimal representation prior to addition to determine the actual measured or calculated Motion Imagery bounding box corner location. Bounding box corners that lie above the horizon do not correspond to points on the earth. Bounding box corners may lie outside of the mapped range. Both cases should either not be reported or be reported as an "error". Conversion: IMAPB(-19.2, 19.2, 3).

<u>Valid Values:</u> The set of real numbers from -19.2 to 19.2 inclusive.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10.00 Degrees	0x10	0x03	IMAPB(-19.2, 19.2, 3, 10.00) = 0x3A66 67

Requirement	
ST 0903.4-35	VTarget Pack Bounding Box Bottom Right Longitude Offset (Tag 16) shall only be present when the VMTI LS is embedded within a MISB ST 0601 LS.

8.2.18 VTarget Pack: Tag 17 – targetLocation

Provides detailed geo-positioning information for a target, optionally including the standard deviation and correlation coefficients. This item is of type Location which is a Defined-Length Truncation Pack. To specify the geographic coordinates for a target when the VMTI LS is not embedded within a MISB ST 0601 LS, Target Location VTarget Pack Tag 17 must be used in lieu of Target Location Latitude Offset VTarget Pack Tag 10 and Target Location Longitude Offset VTarget Pack Tag 11, since ST 0601 Target Latitude and Target Longitude (Frame Center Coordinates) will not be specified. However, even embedding the VMTI LS within a ST 0601 LS, Target Location may still be used. See the Location Type definition for a full description.

Requirement	
ST 0903.4-36	When the VMTI LS is not embedded within a MISB ST 0601 LS, VTarget Pack Target Location (Tag 17) shall be used instead of VTarget Pack Target Location Latitude Offset (Tag 10) and VTarget Pack Target Location Longitude Offset (Tag 11).

8.2.19 VTarget Pack: Tag 18 – targetBoundarySeries

Provides detailed geo-positioning information for the boundary around an area or volume of interest. An arbitrary number of vertices defines the boundary. Each vertex is an element of type Location. Typical boundaries are the bounding boxes defined by two or four vertices. Location type captures geo-positioning data about a specific location on or near the surface of the Earth. The contents of these packs fall into three groups, namely, geospatial location (Latitude, Longitude, and Height), standard deviations for these values, and correlation coefficients among them. Location elements are Defined-Length Truncation Packs, omitting unknown or less important data from the end of the Pack. Use of TargetBoundarySeries is preferred over Target Bounding Box (Tags 13 through 16) when accuracy and correlation information is available and needed. Such information aids fusion with other moving object indicators, such as, radar based GMTI, to support track identification and tracking.

8.2.20 VTarget Pack: Tag 19 – centroidPixRow

Specifies the row of the target centroid within the Motion Imagery frame in pixels. Numbering commences from 1, denoting the top row. May be used with Target Centroid Pixel Column (Tag 20) to provide an alternate method to specify Target Pixel Centroid Number (Tag 1), the pixel location of the target centroid. If present, Target Centroid Pixel Column (Tag 20) must also be present.

<u>Valid Values:</u> Integer values in the range 1 to 2^{32}-1.

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
872	0x13	0x02	0x0368

Requirement	
ST 0903.4-37	Where VTarget Pack Target Centroid Pixel Column (Tag 20) is present, VTarget Pack Target Centroid Pixel Row (Tag 19) shall be present.

8.2.21 VTarget Pack: Tag 20 – centroidPixCol

Specifies the column of the target centroid within the Motion Imagery frame in pixels. Numbering commences from 1, denoting the left column. May be used with Target Centroid Pixel Row (Tag 19) to provide an alternate method to specify Target Pixel Centroid Number (Tag 1), the pixel location of the target centroid. If present, Target Centroid Pixel Row (Tag 19) must also be present.

<u>Valid Values:</u> Integer values in the range 1 to 2^{32}-1.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
1137	0x14	0x02	0x0471

Requirement	
ST 0903.4-38	Where VTarget Pack Target Centroid Pixel Row (Tag 19) is present, VTarget Pack Target Centroid Pixel Column (Tag 20) shall be present,

8.2.22 VTarget Pack: Tag 21 – FpaIndex Pack

Specifies the Focal Plane Array (FPA) in which detection of the target occurs with FPA Row and FPA Column, in that order, in a two-dimensional array of FPAs. The purpose of the FPA Index is to support sensors constructed using multiple FPAs, such as Large Volume Motion Imagery (LVMI). Numbering for Rows is from 1, top to bottom, starting from the top of the array. Numbering for Columns is from 1, left to right, starting from the left of the array.

Table 5 summarizes the FPA Index Defined Length Pack. The column designations are as follows:

- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 5: FPA Index Defined-Length Pack

FPA Index Pack				
Name	**Units**	**Format**	**Len**	**Description**
fpaRow	N/A	uint	1	Specifies the row of a sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs
fpaCol	N/A	uint	1	Specifies the column of a sensor Focal Plane Array (FPA) in a two-dimensional array of FPAs

Valid Values: row and column specified with a value ranging from 1 to 255, inclusive.

Example Value	Example Encoded LS Value		
(row, column) = (2, 3)	Tag	Length	Value
see Figure 11	0x15	0x02	0x0203

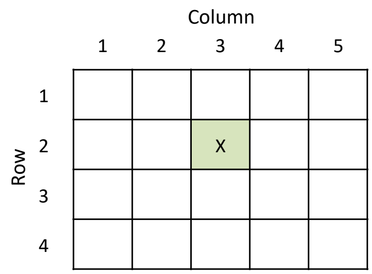

Figure 11: Array of FPAs: "X" located in Row 2, Column 3

Target pixel coordinates are with respect to the FPA of the detected target. View the FPA as a Motion Imagery "Frame" for VMTI data that use the Frame as its reference.

8.2.23 VTarget Pack: Tag 22 – algorithmId

The AlgorithmId is a reference to one of the items in the VMTI Local Set Algorithm Series. The 102_algorithmSeries attribute lists all the algorithms the VMTI Local Set will use. Each algorithm in the series includes an identifier (algorithmId), the AlgorithmId value equals one of the Id values in the Algorithm Series.

8.2.24 VTarget Pack: Tag 101 – vMask

The VMask LS defines the outline of a detected target. Downstream clients can redraw the outline or "chip" the target from the Motion Imagery. Specifying the shape of the outline is by (1) three or more points representing the vertices of a polygon within a Motion Imagery Frame, or (2) a bit mask identifying the pixels within the Motion Imagery Frame subsumed by the target. There is no restriction in specifying both a polygon and a bit mask simultaneously.

8.2.25 VTarget Pack: Tag 102 – vObject

The VObject Local Set describes the class or type of a target (aircraft, watercraft, car, truck, train, dismount, etc.) to an arbitrary level of detail. For example, it might be useful to expand the notion of a "dismount" to include combatant, noncombatant, male, female, etc. This standard mandates the use of the Web Ontology Language (OWL) [10] to define the VObject ontology.

8.2.26 VTarget Pack: Tag 103 – vFeature

Data which describes the target or features of the target (shape, size, dents, number of wheels, thermal signature, etc.). Descriptive information can range from simple text for a label, to complex data sets containing spectral or radiometric data. The definition of a set of elements to describe target features can be a complex undertaking. Rather than create a unique specification, the VFeature LS is based on ISO 19156 [11] and related schemas.

ISO 19156 defines a conceptual schema for observations and for features for sampling during observation. These artifacts support the exchange of information describing the acts of observation and their results. It also defines an observation as an act of measuring or otherwise determining the value for properties of features of interest. Such an act may involve the use of a method, sensor, instrument, human observation, algorithm, computation, process, etc. to estimate the value of the property. An observation is associated with a discrete instant or interval of time.

Values use a variety of scales including nominal, ordinal, ratio, interval, spatial, and temporal. Combining primitive data types forms aggregate data types with aggregate values, including vectors, tensors and images. A value may be exact, or it may be an estimate, with a finite error. Observation results may have many data types, including primitive types like category or measure, but also more complex types such as time, location and geometry.

8.2.27 VTarget Pack: Tag 104 – vTracker

Contains spatial and temporal information ancillary to VChip, VObject, and VFeature to assist in tracking the target. Such information allows Motion Imagery tracking algorithms to produce better tracks from the VMTI target information. Note: In general, use the VTrack (no "er") LS over the VTracker LS for the representation of target tracks.

8.2.28 VTarget Pack: Tag 105 – vChip

Allows inclusion of an image "chip" of the target. This LS will find use in bandwidth constrained environments, where the operator does not have access to the underlying Motion Imagery stream. In general, the image chip will simply be "embedded" with the VMTI metadata. However, this specification permits reference to an image using a Uniform Resource Identifier/Locator (URI / URL) to support linking to a previously stored image, obviating the need to include the image data itself in the stream.

8.2.29 VTarget Pack: Tag 106 – vChipSeries

A Series of one or more VChip LS associated with a specific target as shown in Figure 12. Each LS contains image chip information. When using the VChip LS within a VChipSeries, omit the VChip LS Tag 105 because all values are of the same type. Just specify the Length and Value for each VChip LS represented. Indicating multiple image chips could; for example, an image chip from the source sensor and another image chip from a different sensor, say, one of higher resolution or of a different modality (e.g., IR).

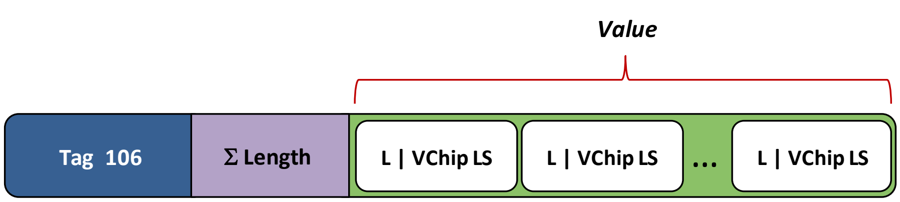

Figure 12: VChipSeries Structure

8.2.30 VTarget Pack: Tag 107 – vObjectSeries

A Series of one or more VObject LS associated with a specific target as shown in Figure 13. Each LS contains ontology information. When using the VObject LS within a VObjectSeries, omit the VObject LS Tag 102 because all values are of the same type. Just specify the Length and Value for each VObject LS represented.

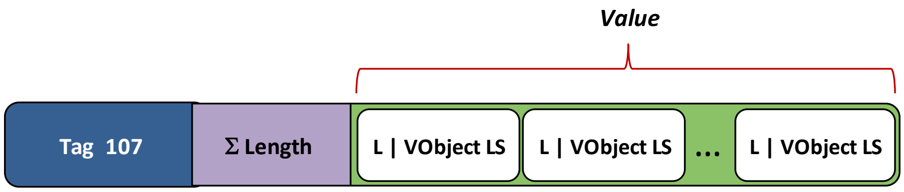

Figure 13: VObject Series Structure

8.3 VMask LS

Table 6 summarizes the VMask Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VMask LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 6: VMask Local Set

VMask Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	polygon	N/A	Array	V	At least three unsigned integer numbers that specify the vertices of a polygon representing the outline of a target
2	bitMaskSeries	N/A	Series	V	Describes the area of the frame occupied by a target using a run-length encoded bit mask with 1 to indicate that a pixel subtends a part of the target and 0 to indicate otherwise

8.3.1 VMask LS: Tag 1 – polygon

Three or more points that represent the vertices of a polygon within a Motion Imagery Frame. List the points in clockwise order. Close the polygon by connecting the last point to the first point. Each point is a pixel number with numbering commencing with 1, at the top left pixel, proceeding from left to right, top to bottom, then encoded using the Length-Value construct of a Variable-Length Pack.

Note: in the UML of the VMTI LS the type of polygon is indicated as an array i.e., [].

The calculation of the pixel number is pixel number = Column + ((Row-1) x frame width)). The top left pixel of the frame equates to (Column, Row) = (1, 1) and pixel number=1. For example, for frame width = 1920, pixel location (1, 1) pixel number=1; pixel location (2, 1) pixel number=2; pixel location (1, 2) pixel number=1921.

Example Value	Example Encoded LS Value		
	Tag	Length	Value (Len, pixel number)
14762, 14783, 15115	0x01	0x09	0x02 39AA 0x02 39BF 0x02 3B0B

Requirement(s)	
ST 0903.4-40	The points specified in VMask LS Polygon (Tag 1) shall be listed in clockwise order.
ST 0903.4-41	Each pixel number specified in VMask LS Polygon (Tag 1) shall be encoded using the Length-Value construct of a Variable-Length Pack.
ST 0903.4-42	The polygon specified in VMask LS Polygon (Tag 1) shall be closed by connecting the last point to the first point.
ST 0903.5-100	A polygon pixel number shall be no greater than the number of pixels in a frame.

8.3.2 VMask LS: Tag 2 – bitMaskSeries

A run-length encoding as a Series Type of a bit mask describing the pixels that subtend the target within the Motion Imagery Frame. Pixel-number-plus-run-length pairs, each describing the starting pixel number and the number of pixels in a run (see Figure 14). Pixel numbering commences with 1, at the top left pixel, proceeding from left to right, top to bottom. Encode pixel numbers using the Length-Value construct of a Variable-Length Pack. Encode the length of each run using BER-Length encoding. The criterion used to decide whether a pixel "covers" a portion of the target is somewhat arbitrary and left to the implementer. The implementer is free to decide whether overlap with all, a majority, or just a fraction of the pixel constitutes "covering" the target.

Requirement	
ST 0903.5-101	A run length shall be contained within an image frame.

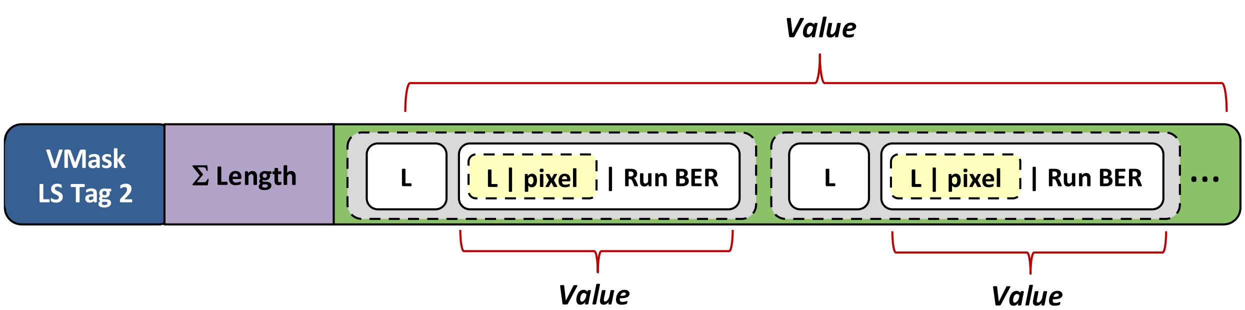

Figure 14: Series structure for encoding pixel runs

The calculation of the pixel number is pixel number Column + ((Row-1) x frame width)). The top left pixel of the frame equates to (Column, Row) = (1, 1) with pixel number=1, then encoded using the Length-Value construct of a Variable-Length Pack.

For example, in Figure 15 the pairs of pixel numbers and run lengths (pixel, run) are:

(74, 2) = [0x01 4A] [0x02]   (89, 4) = [0x01 59] [0x04]   (106, 2) = [0x01 6A] [0x02]

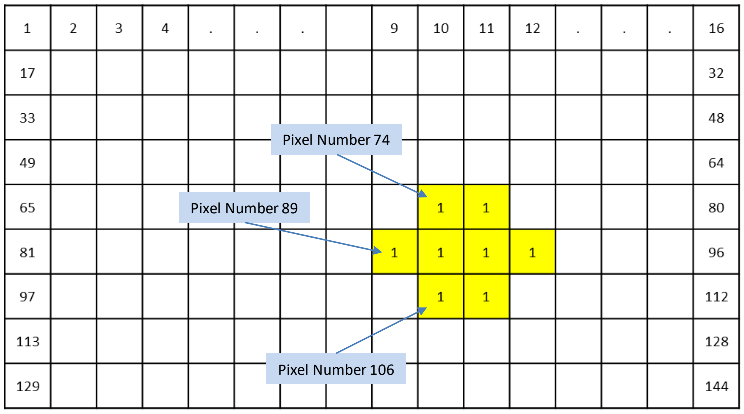

Figure 15: Example Bit Mask

In this example, each run length is a single byte using the short form of BER-Length encoding. The long form of BER-Length encoding is for run lengths exceeding 127 pixels. Run lengths typically (but not necessarily) will be "small" leveraging the bit efficiency of the short form.

Example Value	Example Encoded LS Value		
	Tag	Length	Value (Len, Len, Pixel Num (BER-Len), Run)
(pixel, run) = (74, 2), (89, 4), (106, 2)	0x02	0x0F	0x04, 0x02, 0x014A, 0x02 0x04, 0x02, 0x0159, 0x04 0x04, 0x02, 0x016A, 0x02

Requirement(s)	
ST 0903.4-43	Each run of pixels in VMask LS Bit Mask (Tag 2) that subtends a part of the target shall be encoded by specifying the number of the pixel at the start of the run and the number of pixels in the run.
ST 0903.4-44	The run length in VMask LS Bit Mask (Tag 2) shall be encoded using BER-Length encoding.

8.4 VObject LS

Table 7 summarizes the VObject Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VObject LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 7: VObject Local Set

VObject Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	ontology	N/A	utf8	V	Uniform Resource Identifier (URI) which refers to a VObject ontology
2	ontologyClass	N/A	utf8	V	The name of the target class or type, as defined in the VObject Ontology
3	ontologyId	N/A	uint	V3	Identifier indicating which Ontology in the VMTI's Ontology Series represents this object
4	confidence	N/A	IMAP	V3	The amount of confidence in the classification of this object

8.4.1 VObject: Tag 1 – ontology

The Ontology Tag 1 item is a Uniform Resource Identifier (URI), which refers to a VObject ontology. Refer to the OWL Web Ontology Language for the ontology used. The Jet Propulsion Laboratory Semantic Web for Earth and Environmental Terminology (SWEET) (https://bioportal.bioontology.org/ontologies/SWEET) provides a collection of ontologies, written in the OWL ontology language which serves as examples and starting points for the development of additional domain-specific extended ontologies.

The Ontology Tag 1 needs to precede the ontologyClass Tag 2 item. For bandwidth efficiency, Ontology can appear periodically and referenced by subsequent VTarget Packs which use the specified ontology.

Requirement(s)	
ST 0903.4-45	The ontology referred to by the URI of the VObject LS Ontology (Tag 1) item shall be expressed using the OWL Web Ontology Language.
ST 0903.4-46	The VObject LS Ontology (Tag 1) item shall appear prior to any appearance of the OntologyClass (Tag 2).
ST 0903.4-47	The VObject Ontology (Tag 1) item shall appear from time to time in the KLV stream as an item with Periodic Volatility.

8.4.2 VObject: Tag 2 – ontologyClass

A value representing a target class or type, as defined by the VObject Ontology Tag 1. For bandwidth efficiency, it is desirable that the Ontology specify a mapping between compact values (perhaps BER-OID encoded) for use in OntologyClass and more descriptive names for use by systems that present the information to human observers.

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
Dismount/Non-combatant/Female/Child	0x02	0x23	0x4469 736D 6F75 6E74 2F4E 6F6E 2D63 6F6D 6261 7461 6E74 2F46 656D 616C 652F 4368 696C 64

Requirement	
ST 0903.4-48	The VObject LS OntologyClass (Tag 2) shall have a value taken from the ontology specified by VObject LS Ontology (Tag 1) that precedes it in the KLV stream.

8.4.3 VObject: Tag 3 – ontologyId

The OntologyId is an alternative method for specifying the information in Tag 1 (Ontology) and Tag 2 (OntologyClass) when the VMTI LS includes an Ontology Series in Tag 103. The OntologyId is a reference to one of the elements in the VMTI LS Ontology Series. The 103_ontologySeries attribute lists all the ontologies the VMTI LS will use. Each ontology in the series includes an identifier (Id), the OntologyId value equals one of the Id values in the Ontology Series. Using the OntologyId replaces the need to use Tag 1 and Tag 2 in VObject LS saving bandwidth by not duplicating the same information for different VObjects.

8.4.4 VObject: Tag 4 – confidence

The Confidence value is the measure of "trust" in the classification of this VObject, ranging from 0.0% to 100.0%. For example, an object classifier will analyze a blob of pixels and "guess" the blob represents a vehicle in the image; if the blob is well defined and matches the classifiers criteria for a vehicle very closely the confidence in the classification is high (towards 100.0%), alternatively if the classifier is less sure of its classification the confidence is low (towards 0%). The confidence value is IMAPB(0.0, 100.0, length) with the length defined by the tag's length value. Increasing the length provides more accuracy, the minimum length is 1.

8.5 VFeature LS

Table 8 summarizes the VFeature Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VFeature LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 8: VFeature Local Set

VFeature Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	schema	N/A	utf8	V	Uniform Resource Identifier (URI) which refers to an OGC Geography Markup Language (GML) Observations and Measurements (O&M) application schema
2	schemaFeature	N/A	utf8	V	OGC GML document structured according to the schema specified by Tag 1. Intended to capture properties (values) observed for a feature of interest.

8.5.1 VFeature LS: Tag 1 – schema

A Uniform Resource Identifier (URI) which points to a relevant Observation schema (http://schemas.opengis.net/om/1.0.0/) or a related schema. The O&M standard defines an Observation as "an action with a result which has a value describing some phenomenon". The Observation is a Feature within the context of ISO 19101 [12] and ISO 19109 [13]. An Observation may involve use of a sensor or observer, analytical procedure, simulation, or other mathematical process. An Observation yields an estimate of the value of a property of the feature of interest and can account for error that may be present in the estimate.

Observation values may have many data types, from primitive to complex, including category, measure, and geometry. Combinations of data types can encode properties having multiple components. The notion of "coverage" is useful for properties that vary over the feature of interest. O&M Sampling addresses the sampling of sub-elements of a feature used to represent the whole.

Schema is a required element of VFeature. However, for bandwidth efficiency since Scheme appears periodically, SchemaFeature items in subsequent VTarget Packs reference the specified schema.

Requirement(s)	
ST 0903.4-49	The VFeature LS shall conform to ISO 19156 and related schemas.
ST 0903.4-50	The VFeature LS Schema (Tag 1) item shall be present prior to the appearance of a VFeature LS Feature (Tag 2) item.
ST 0903.4-51	The VFeature Schema (Tag 1) item shall appear from time to time in the KLV stream as an item with Periodic Volatility.

8.5.2 VFeature LS Tag 2 – schemaFeature

A Geographic Markup Language (GML) document structured according to the schema specified by VFeature LS Tag 1 Schema. It may contain one or more values observed for a feature of interest.

Valid Values: Any OGC GML document that validates against the schema specified in Tag 1: Schema

Example Value:

(Temperature, Pressure) =

```
<gml:DataBlock>
  <gml:rangeParameters>
    <gml:CompositeValue>
      <gml:valueComponents>
        <Temperature uom="urn:x-si:v1999:uom:degreesC">template</Temperature>
        <Pressure uom="urn:x-si:v1999:uom:kPa">template</Pressure>
      </gml:valueComponents>
    </gml:CompositeValue>
  </gml:rangeParameters>
  <gml:tupleList>3,101.2</gml:tupleList>
```

</gml:DataBlock>

Example Value	Example Encoded LS Value		
	Tag	Length	Value
<gml:DataBlock> … </gml:DataBlock> (see Example Value)	0x02	0x8201 43	0x3C67 6D6C 3A44 6174 6142 6C6F 636B 3E3C 676D 6C3A 7261 6E67 … … 6C65 4C69 7374 3E3C 2F67 6D6C 3A44 6174 6142 6C6F 636B 3E

Requirement	
ST 0903.4-52	The VFeature LS SchemaFeature (Tag 2) shall conform to the schema specified by VFeature LS Schema (Tag 1).

8.6 VTracker LS

Table 9 summarizes the VTracker Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VTracker LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g.,µs is micro-seconds)
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 9: VTracker Local Set

VTracker Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	trackId	N/A	uint	16	A unique identifier (UUID) for the track
2	detectionStatus	N/A	uint	1	Enumeration indicating the current state of VMTI detections for a given entity (Inactive, Active, Dropped, Stopped)
3	startTime	µs	uint	8	Start time for the first observation of the entity
4	endTime	µs	uint	8	End time for the most recent observation of the entity
5	boundarySeries	N/A	Series	V	Set of vertices of type Location that specify a minimum bounding area or volume, which encloses the full extent of VMTI detections for the entity
6	algorithm	N/A	utf8	V	Name or description of the algorithm or method used to create or maintain object movement reports or intervening predictions of such movement

VTracker Local Set					
Tag ID	Name	Units	Format	Len	Description
7	confidenceLevel	N/A	uint	1	An estimation of the certainty or correctness of VMTI movement detections. Larger values indicate greater confidence.
8	numTrackPoints	N/A	uint	V2	Number of coordinates of type Location that describe the history of VMTI detections described by trackHistorySeries
9	trackHistorySeries	N/A	Series	V	Points of type Location that represent the locations of VMTI detections
10	velocity	N/A	Velocity	V	Velocity of the entity at the time of last observation
11	acceleration	N/A	Acceleration	V	Acceleration of the entity at the time of last observation
12	algorithmId	N/A	uint	V3	Identifier indicating which algorithm in the Algorithm Series tracked this target

8.6.1 VTracker LS: Tag 1 – trackId

Uniquely identifies a track, using a 128-bit (16-byte) Universal Unique Identification (UUID) as standardized by the Open Software Foundation according to ISO/IEC 9834-8 [14].

Example Value	Example Encoded LS Value		
f81d4fae-7dec-11d0-a765-00a0c91e6bf6	Tag	Length	Value
	0x01	0x10	0xF81D 4FAE 7DEC 11D0 A765 00A0 C91E 6BF6

Requirement	
ST 0903.4-53	The VTracker LS Track ID (Tag 1) shall be a 16-byte Universal Unique Identification (UUID) in accordance with ISO/IEC 9834-8.

8.6.2 VTracker LS: Tag 2 – detectionStatus

An enumerated data value with values assigned from Table 10[5].

Table 10: Detection Status Values

Value	Status	Description
0	Inactive	The VMTI detections for the entity have ended. The entity may have merged with one or more other entities; have split into two or more new entities; or have ceased to exist because no VMTI detection can be correlated with it.

[5] Ideally (eventually), the Track Status Values and those of NATO STANAG 4676 Tracking Standard for ISR Systems should be defined consistently.

1	Active	Detections for the entity established or updated based on associated VMTI report or prediction. An entity can resume this state by transition from Stopped or Dropped to "moving" when a VMTI detection (or a prediction) with a new position has become associated with it.
2	Dropped	The entity could not be correlated with any VMTI detection for an interval of time exceeding a specified threshold. An entity can remain in a Dropped or "lost" condition for an indeterminate period if there is some likelihood it may resume (Active) again. Eventually, it may become Inactive.
3	Stopped	The entity has either become stationary or was always in a fixed location.

8.6.3 VTracker LS: Tag 3 – startTime

Captures the time of the first observation of the entity in microseconds elapsed since midnight (00:00:00), January 1, 1970 (see MISB ST 0603).

Example Value	Example Encoded LS Value		
	Tag	Length	Value
April 19, 2001, 04:25:21 GMT (987654321000000)	0x03	0x07	0x0382 4430 F6CE 40

8.6.4 VTracker LS: Tag 4 – endTime

Captures the time of the most recent observation of the entity in microseconds elapsed since midnight (00:00:00), January 1, 1970 (see MISB ST 0603).

8.6.5 VTracker LS: Tag 5 – boundarySeries

BoundarySeries is of type BoundarySeries that specifies a bounding area or volume, which encloses the full extent of VMTI detections for the entity. For a simple, planar bounding box, the area will generally lie on the surface of the Earth (although not necessarily, depending upon the Height values provided) defining the "footprint" of the track. By specifying additional vertices, enables describing complex, multifaceted volumes.

Requirement	
ST 0903.4-54	VTracker LS BoundarySeries (Tag 5) vertices shall be ordered so that looking toward Earth center they spiral in a clockwise direction from lowest elevation to highest.

8.6.6 VTracker LS: Tag 6 – algorithm

Unique name or description of the algorithm or method used to create or maintain object movement reports or intervening predictions of such movement.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
test	0x06	0x04	0x7465 7374

8.6.7 VTracker LS: Tag 7 – confidenceLevel

The confidenceLevel, expressed as a percentage from 0 to 100, is an estimation of the certainty or correctness that the track described by the sequence of VMTI movement detections corresponds to the same object. Value 0 indicates no confidence; value 100 percent indicates absolute certainty. Confidence is an estimation of the certainty or correctness that the track described by the sequence of VMTI movement detections corresponds to the same object. For example, detections derived from many unambiguous target reports, such as, for a single vehicle on a road in a desert environment might signal high confidence. Reports associated with several overlapping or nearby tracks in a partially obscured environment, such as, for dismounts (people) in an urban setting might signal low confidence.

Valid values: The set of all integers from 0 to 100 inclusive.

Example Value	Example Encoded LS Value		
50%	Tag	Length	Value
	0x07	0x01	0x32

8.6.8 VTracker LS: Tag 8 – numTrackPoints

The number of coordinates of type Location that describe the trackHistorySeries of VMTI detections for the target. Strictly speaking, Number of Track Points is unnecessary, since the value is derivable from the Length information associated with VTracker Tag 9 trackHistorySeries. If specified, the number of track points needs to be at least 1; else no trackHistorySeries would exist.

Valid Values: The set of all integers from 1 to 65,536 inclusive.

Example Value	Example Encoded LS Value		
27	Tag	Length	Value
	0x08	0x01	0x1B

Requirement	
ST 0903.4-55	Where an instance of the VTracker LS is present, the number of track points specified by VTracker LS Number of Track Points (Tag 8) shall be at least one (1).

8.6.9 VTracker LS: Tag 9 – trackHistorySeries

A Series of points that represent the locations of entity VMTI detections. Each point is an element of type Location. Points are chronologically in order from start to end of the VMTI detections.

Requirement	
ST 0903.4-56	Points in a trackHistorySeries shall be ordered chronologically from start to end of the VMTI detections.

8.6.10 VTracker LS: Tag 10 – velocity

The velocity of the entity at the time of last detection.

Valid Values: See Velocity Type (Section 8.10.2)

8.6.11 VTracker LS: Tag 11 – acceleration

The acceleration of the entity at the time of last detection.

Valid Values: See Acceleration Type (Section 8.10.3)

8.6.12 VTracker LS: Tag 12 – algorithmId

Specifies identifier number assigned to detection algorithm used.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
3	0x0B	0x01	0x03

8.7 VChip LS

Table 11 summarizes the VChip Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VChip LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 11: VChip Local Set

Local Set Key					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	imageType	N/A	utf8	V	Internet Assigned Numbers Authority (IANA) image media subtype specifying the VChip image type (limited to "jpeg", and "png")
2	imageUri	N/A	utf8	V	Uniform Resource Identifier (or Uniform Resource Locator) that refers to an image stored on a server
3	embeddedImage	N/A	binary	V	An image "chip" of the image type specified by Tag 1

8.7.1 VChip LS: Tag 1 – imageType

A string of UTF-8 characters that correspond to an IANA media image subtype. Only the IANA media image subtypes "jpeg" and "png" are allowed; these are common formats for compressing a still image. JPEG is a lossy compression method, but quality is adjustable. PNG is lossless and provides RGB bit depths up to 48 bits per pixel (16 bits per color component). Thus, it preserves "raw" pixel values.

Tag 1 Image Type is a required item in the VChip LS. However, for bandwidth efficiency, Image Type is not necessary in every VChip LS (although it may) but needs to be present periodically. Once specified Image Type, Image URI and Embedded Image items in subsequent VChip LSs needs to be consistent with the specified image type.

Valid Values: A string of UTF-8 characters that correspond to an IANA media image subtype.

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
jpeg	0x01	0x04	0x6A70 6567

Requirement(s)	
ST 0903.4-57	The VChip LS Image Type (Tag 1) item shall appear in the KLV stream prior to a VChip LS Image URI (Tag 2) or a VChip LS Embedded Image (Tag 3) item.
ST 0903.4-58	The VChip LS Image Type (Tag 1) item shall appear in the KLV stream with Periodic Volatility.
ST 0903.4-59	The type of the image referred to by VChip LS Image URI (Tag 2) shall be that specified by the preceding VChip LS Image Type (Tag 1).
ST 0903.4-60	The type of a VChip LS Embedded Image (Tag 3) shall be that specified by the preceding VChip LS Image Type (Tag 1).

8.7.2 VChip LS: Tag 2 – imageUri

A Uniform Resource Identifier (usually, a Uniform Resource Locator) that refers to an image of the type specified by VChip LS Image Type Tag 1, stored on a network or a file system. In some situations, probably downstream from the collection source, such a reference could be used in lieu of embedding the image chip in the stream.

Valid Values: A string of UTF-8 characters that comply with the rules for building a valid URI.

8.7.3 VChip LS: Tag 3 – embeddedImage

An image "chip" of the type specified by VChip Image Type Tag 1, embedded in the VMTI stream.

Valid Values: Any image implemented in compliance with the IANA media image subtype specified in Tag 1.

8.8 Algorithm LS

The Algorithm LS documents attributes of the algorithm used for detection and tracking of targets. The AlgorithmSeries Tag 102 conveys one or more instances of the LS allowing for documenting different algorithms in use within one VMTI LS.

Table 12 summarizes the Algorithm LS. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the Algorithm LS
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 12: Algorithm Local Set

Algorithm Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	id	N/A	uint	V3	Identifier for the algorithm used (Do not use 0x00, it is reserved for future use)
2	name	N/A	utf8	V	Name of algorithm
3	version	N/A	utf8	V	Version of algorithm
4	class	N/A	utf8	V	Type of algorithm e.g., detector classifier
5	nFrames	N/A	uint	V	Number of frames the algorithm operates over

8.8.1 Algorithm LS: Tag 1 – id

When the VMTI LS contains a Series of Algorithm Local Sets each element in the Series needs a unique identifier (ID). The ID is an integer which identifies a single Algorithm LS in the Series and is unique among all elements in the list. Systems may reuse IDs from VMTI LS to VMTI LS (i.e., two sequential VMTI packets) so receivers should not assume an identifier value is static for an entire VMTI stream. The ID does not need to start at a value of one (1) nor do the IDs need to be in any specific order in the AlgorithmSeries.

Requirement	
ST 0903.5-102	All instances of the Algorithm LS shall contain Tag 1, id.

Valid Values: An unsigned integer greater than 0x00 (Value 0x00 is Reserved).

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
9	0x01	0x01	0x09

8.8.2 Algorithm LS: Tag 2 – name

The Name is the name assigned to the algorithm by the data producer.

Requirement	
ST 0903.5-103	All instances of the Algorithm LS shall contain Tag 2 name.

Valid Values: Any alphanumeric value in UTF8.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
k6_yolo_9000_tracker	0x02	0x14	0x6B36 5F79 6F6C 6F5F 3930 3030 5F74 7261 636B 6572

8.8.3 Algorithm LS: Tag 3 – version

The Version is the version of the algorithm.

Requirement	
ST 0903.5-104	All instances of the Algorithm LS shall contain Tag 3 version.

Valid Values: Any alphanumeric value in UTF8.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
2.6a	0x03	0x04	0x322E 3661

8.8.4 Algorithm LS: Tag 4 – class

The Class is the type of algorithm.

Valid Values: Any alphanumeric value in UTF8.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
kalmann	0x04	0x07	0x6B61 6C6D 616E 6E

8.8.5 Algorithm LS: Tag 5 – nFrames

The nFrames is the number of frames the algorithm processes when detecting or tracking the object.

Example Value	Example Encoded LS Value		
	Tag	Length	Value
10	0x05	0x01	0x0A

8.9 Ontology LS

The Ontology LS describes the class or type of a target (aircraft, watercraft, car, truck, train, dismount, etc.) to an arbitrary level of detail. For example, it might be useful to expand the notion of a “dismount” to include combatant, noncombatant, male, female, etc. This standard mandates the use of the Web Ontology Language (OWL) [10] to define the ontology. The Ontology LS contains four items, Tag 1 Ontology and Tag 2 Class, both required to be present.

Table 13 summarizes the Ontology LS. The column designations are as follows:

- “Tag ID” indicates the Tag to use when specifying the item in the Ontology LS

- “Name” is the label associated with the tag
- “Units” specifies the units of the data specified in the Value field
    - Units of “N/A” indicates no units apply
- “Format” indicates the item’s format for the Value
- “Len” indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 13: Ontology Local Set

Ontology Local Set					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	id	N/A	uint	V3	Identifier for the ontology used
2	parentId	N/A	uint	V3	Defines the link when an OntologySeries has two related LS in the Series
3	ontology	N/A	utf8	V	Uniform Resource Identifier (URI) according to the OWL Web Ontology Language
4	ontologyClass	N/A	utf8	V	A value representing a target class or type, as defined by the Ontology

8.9.1 Ontology LS: Tag 1 – id

When the VMTI LS contains a Series of Ontology Local Sets each element in the Series needs a unique identifier. The Id is an integer which identifies a single Ontology LS in the Series and is unique among all elements in the list. Systems may reuse Ids from VMTI LS to VMTI LS (i.e., two sequential VMTI packets) so receivers should not assume an identifier value is static for a whole VMTI stream. The Id does not need to start at a value of one nor do the Ids need to be in any specific order in the Ontology Series.

Requirement	
ST 0903.5-105	All instances of the Ontology LS shall contain Tag 1 id.

8.9.2 Ontology LS: Tag 2 – parentId

When detecting or tracking objects there may be several related classifications made for an object. For example, a blob of pixels moving with a high velocity may classify as both a vehicle (a generalization) with a confidence level of 90%, a car with a confidence level of 70%, and a motorcycle with a confidence level of 50%. Vehicle is a generalization of both car and motorcycle; this relationship between car and vehicle, and motorcycle and vehicle is indicated with the parentId.

When an OntologySeries has two related LS in the Series, the ParentId defines the link by having the child define its parentId equal to the parent ontology object’s OntologyId. For example, an OntologySeries having three elements: Vehicle with OntologyId 10, Car with OntologyId 17, and Motorcycle with OntologyId 3. Since Car and Motorcycle are both “children” of the Vehicle, those two LS define their ParentId’s equal to 10.

8.9.3 Ontology LS: Tag 3 – ontology

The Ontology Tag 3 item is a Uniform Resource Identifier (URI). Refer to the OWL Web Ontology Language for the ontology used. The Jet Propulsion Laboratory Semantic Web for Earth and Environmental Terminology (SWEET) (https://bioportal.bioontology.org/ontologies/SWEET) provides a collection of ontologies, written in the OWL ontology language which serves as examples and starting points for the development of additional domain-specific extended ontologies.

The Ontology Tag 3 needs to precede the OntologyClass Tag 4 item. For bandwidth efficiency, Ontology can appear periodically.

Requirement(s)	
ST 0903.5-106	The ontology referred to by the URI of the Ontology LS Ontology (Tag 3) item shall be expressed using the OWL Web Ontology Language.
ST 0903.5-107	All instances of the Ontology LS shall contain Tag 3 ontology.

8.9.4 Ontology LS: Tag 4 – ontologyClass

A value representing a target class or type, as defined by the Ontology LS Tag 3 Ontology. For bandwidth efficiency, it is desirable that the Ontology specify a mapping between compact values (perhaps BER-OID encoded) for use in OntologyClass and more descriptive names for use by systems that present the information to human observers.

Requirement	
ST 0903.5-108	All instances of the Ontology LS shall contain Tag 4 ontologyClass.

Example Value	**Example Encoded LS Value**		
	Tag	Length	Value
Dismount/Non-combatant/Female/Child	0x04	0x23	0x4469 736D 6F75 6E74 2F4E 6F6E 2D63 6F6D 6261 7461 6E74 2F46 656D 616C 652F 4368 696C 64

8.10 Location, Velocity, Acceleration and BoundarySeries

The Location, Velocity, Acceleration and Boundary are Defined-Length Pack (DLP) KLV structures.

Requirement(s)	
ST 0903.4-62	Truncation of Location, Velocity, and Acceleration Defined-Length Truncation Packs shall be allowed only at a group boundary.
ST 0903.4-63	Within a Location, Velocity, or Acceleration Defined-Length Truncation Pack, no filler values shall be used for (unknown) higher priority elements.

8.10.1 Location DLP

The Location DLP structure captures geo-positioning data about a specific location on or near the surface of the Earth. The DLP is a truncation pack where groups of data are optional unless needed (truncated from the end only). There are three defined groups as shown in Figure 16.

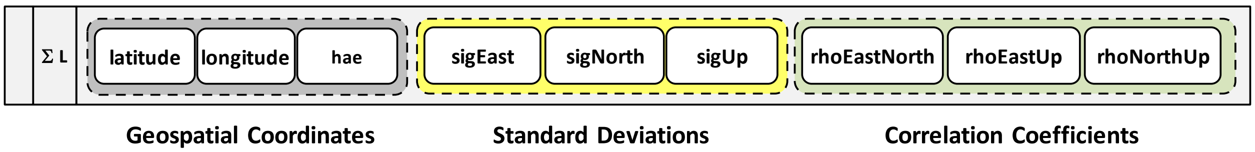

Figure 16: Location DLP Structure

The first, and highest priority group, includes Latitude, Longitude, and Height Above Ellipsoid (HAE). These values define the origin of a local tangential East-North-Up (ENU) coordinate system. The second group Standard Deviations and the third group Correlation Coefficients express uncertainty with respect to the ENU coordinate axes. The Standard Deviations is a medium priority group providing standard deviations for the location in meters (with respect to the ENU coordinate axes). The Correlation Coefficients is the lowest priority group providing correlation coefficients for the location (with respect to the ENU coordinate axes). The correlation coefficients provide a measure of systematic behavior, whether variation in the values of pairs of variables is "coupled" or random.[6]

Standard deviations and correlation coefficients express confidence in the geo-coordinates. For example, if standard deviations are small and correlation coefficients are near unity, then "confidence" in the accuracy of the coordinate values is high. Conversely, if standard deviations are large and correlation coefficients are near zero, potentially large, random errors are likely.

Table 14 summarizes the Location DLP. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the Location DLP
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g., m is meters, ° is degrees)
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

[6] Correlation is dimensionless with a fixed range of values from -1.0 to +1.0, inclusive. Covariance, while it is a similar measure of "relatedness," is in units obtained by multiplying the units of two variables, and thus has values less well constrained.

Table 14: Location DLP Truncation Pack

Location DLP Truncation Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	latitude	°	IMAPB(-90, 90, 4)	4	Latitude in degrees of a point with respect to the WGS84 datum
N/A	longitude	°	IMAPB(-180, 180, 4)	4	Longitude in degrees of a point with respect to the WGS84 datum
N/A	hae	m	IMAPB(-900, 19000, 2)	2	Height of a point in meters above the WGS84 Ellipsoid (HAE)
N/A	sigEast	m	IMAPB(0, 650, 2)	2	Standard deviation of the location of the point with respect to the ENU coordinate system East axis
N/A	sigNorth	m	IMAPB(0, 650, 2)	2	Standard deviation of the location of the point with respect to the ENU coordinate system North axis
N/A	sigUp	m	IMAPB(0, 650, 2)	2	Standard deviation of the location of the point with respect to the ENU coordinate system Up axis
N/A	rhoEastNorth	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between the East and North components of error
N/A	rhoEastUp	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between East and Up components of error
N/A	rhoNorthUp	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between North and Up components of error

Requirement(s)	
ST 0903.4-65	The Location Pack shall consist of up to three groups of information that include a Geospatial Coordinates triplet, a Standard Deviations triplet, and a Correlation Coefficients triplet, in that order.
ST 0903.4-66	The Geospatial Coordinates triplet of a Location Pack shall always be present.
ST 0903.4-67	The Geospatial Coordinates triplet of a Location Pack shall consist of values for Latitude, Longitude, and Height, in that order.
ST 0903.4-68	The Latitude, Longitude, and HAE values of the Geospatial Coordinates triplet of the Location Pack all shall use the WGS84 Ellipsoid as reference.
ST 0903.4-69	The HAE value in the Geospatial Coordinates triplet of the Location Pack shall be expressed as Height Above the Ellipsoid (HAE) in meters with respect to the WGS84 ellipsoid.
ST 0903.4-70	Where the Correlation Coefficients triplet of a Location data type is present, the Standard Deviations triplet shall also be present.
ST 0903.4-71	The Standard Deviations triplet of the Location Pack shall consist of values for the standard deviations of the values in the Geospatial Coordinates triplet with respect to the East-North-Up local coordinate system, specifically East, North, and Up, in that order.

ST 0903.4-72	The Correlation Coefficients triplet of the Location Pack shall consist of values for the pairwise correlation coefficients of the values in the Geospatial Coordinates triplet with respect to the East-North-Up local coordinate system, specifically East-to-North, East-to-Up, and North-to-Up, in that order.

8.10.2 Velocity DLP

The Velocity DLP structure captures data about the velocity of a moving object. The DLP is a truncation pack where groups of data are optional unless needed (truncated from the end only). There are three defined groups as shown in Figure 17. The first, and highest priority group includes East, North and Up velocity components; these provide the measurements of velocity along the coordinate axes of the East-North-Up coordinate system specified by the Location Truncation Pack for the location of the moving object. The second group, Standard Deviations and medium priority group, provides standard deviations for the first group measurements. The third group, Correlation Coefficients and lowest priority group, provides three correlation coefficients for values in the first group.

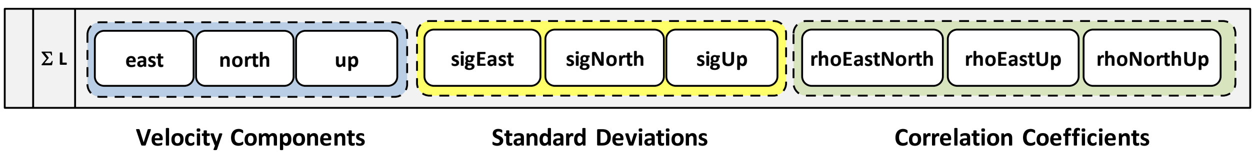

Figure 17: Velocity DLP Structure

Table 15 summarizes the Velocity DLP. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the Velocity DLP
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g., m/s is meters per second)
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 15: Velocity DLP Truncation Pack

Velocity DLP Truncation Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	east	m/s	IMAPB(-900, 900, 2)	2	Velocity along the East axis of the East-North-Up coordinate system
N/A	north	m/s	IMAPB(-900, 900, 2)	2	Velocity along the North axis of the East-North-Up coordinate system
N/A	up	m/s	IMAPB(-900, 900, 2)	2	Velocity along the Up axis of the East-North-Up coordinate system
N/A	sigEast	m/s	IMAPB(0, 650, 2)	2	Standard deviation along East axis
N/A	sigNorth	m/s	IMAPB(0, 650, 2)	2	Standard deviation along North axis

Velocity DLP Truncation Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	sigUp	m/s	IMAPB(0, 650, 2)	2	Standard deviation along Up axis
N/A	rhoEastNorth	m/s	IMAPB(-1, 1, 2)	2	Correlation coefficient between East and North
N/A	rhoEastUp	m/s	IMAPB(-1, 1, 2)	2	Correlation coefficient between East and Up
N/A	rhoNorthUp	m/s	IMAPB(-1, 1, 2)	2	Correlation coefficient between North and Up

Requirement(s)	
ST 0903.5-109	The Velocity Pack shall consist of up to three groups of information that include a Velocity Components triplet, a Standard Deviations triplet, and a Correlation Coefficients triplet, in that order.
ST 0903.4-75	The Velocity Components triplet of the Velocity pack shall be present.
ST 0903.4-76	The Velocity Components triplet of the Velocity pack shall consist of values for East, North, and Up, in that order.
ST 0903.4-77	The East, North, and Up values of the Velocity Components triplet of the Velocity pack shall be expressed using the East-North-Up coordinate system specified by a Location pack for the location of the moving object.
ST 0903.4-78	The East, North, and Up values in the Velocity Components triplet of the Velocity pack shall be expressed in meters per second.
ST 0903.4-79	When the Velocity pack Correlation Coefficients triplet is present, the Velocity pack Standard Deviations triplet shall also be present.
ST 0903.4-80	The Standard Deviations triplet of the Velocity Pack shall consist of values for the standard deviations of the East, North, and Up values in the Velocity Components triplet, in that order.
ST 0903.4-81	The Correlation Coefficients triplet of the Velocity pack shall consist of values for the pairwise correlation coefficients of the values in the Velocity Components triplet, specifically East-to-North, East-to-Up, and North-to-Up, in that order.
ST 0903.4-82	The Velocity pack shall be present only if there is an associated Location pack.

8.10.3 Acceleration DLP

The Acceleration DLP structure captures data about the acceleration of a moving object. The DLP is a truncation pack where groups of data are optional unless needed (from the end only). There are three defined groups as shown in Figure 18. The first, and highest priority group includes East, North and Up components; these measurements provide acceleration along the coordinate axes of the East-North-Up coordinate system specified by the Location pack for the location of the moving object. The second group, Standard Deviations and medium priority group, provides the standard deviations for the first group measurements. The third group, Correlation Coefficients and lowest priority group, provides the three correlation coefficients for the values in the first group.

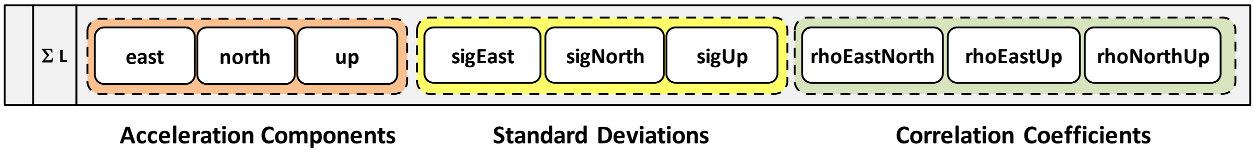

Figure 18: Acceleration Structure

Table 16 summarizes the Acceleration DLP. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the Acceleration DLP
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
    - All other Units are SI enumerations (e.g., m/s^2 is meters per second squared)
- "Format" indicates the item's format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.

Table 16: Acceleration DLP Truncation Pack

Acceleration DLP Truncation Pack					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	east	m/s^2	IMAPB(-900, 900, 2)	2	Acceleration along the East axis of the East-North-Up coordinate system
N/A	north	m/s^2	IMAPB(-900, 900, 2)	2	Acceleration along the North axis of the East-North-Up coordinate system
N/A	up	m/s^2	IMAPB(-900, 900, 2)	2	Acceleration along the Up axis of the East-North-Up coordinate system
N/A	sigEast	m/s^2	IMAPB(0, 650, 2)	2	Standard deviation along East axis
N/A	sigNorth	m/s^2	IMAPB(0, 650, 2)	2	Standard deviation along North axis
N/A	sigUp	m/s^2	IMAPB(0, 650, 2)	2	Standard deviation along Up axis
N/A	rhoEastNorth	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between East and North
N/A	rhoEastUp	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between East and Up
N/A	rhoNorthUp	N/A	IMAPB(-1, 1, 2)	2	Correlation coefficient between North and Up

Requirement(s)	
ST 0903.4-84	The Acceleration pack shall consist of up to three groups of information that include an Acceleration Components triplet, a Standard Deviations triplet, and a Correlation Coefficients triplet, in that order.
ST 0903.4-85	The Acceleration Components triplet of the Acceleration pack shall always be present.
ST 0903.4-86	The Acceleration Components triplet of the Acceleration pack shall consist of values for East, North, and Up, in that order.

ST 0903.4-87	The East, North, and Up values in the Acceleration Components triplet of the Acceleration pack shall be expressed in meters per second squared.
ST 0903.4-88	The East, North, and Up values of the Acceleration Components triplet of the Acceleration pack shall be expressed using the East-North-Up coordinate system specified by a Location pack for the location of the moving object.
ST 0903.4-89	When the Acceleration pack Correlation Coefficients triplet is present, the Acceleration pack Standard Deviation triplet shall also be present.
ST 0903.4-90	The Standard Deviations triplet of the Acceleration pack shall consist of values for the standard deviations of the East, North, and Up values in the Acceleration Components triplet, in that order.
ST 0903.4-91	The Correlation Coefficients triplet of the Acceleration pack shall consist of values for the pairwise correlation coefficients of the values in the Acceleration Components triplet, specifically East-to-North, East-to-Up, and North-to-Up, in that order.
ST 0903.4-92	The Acceleration pack shall be present only if there is an associated Location pack.

8.10.4 BoundarySeries

The BoundarySeries is a Series of Location Packs that define an area or volume enclosing a region of interest. The Boundary structure includes the BER-encoded Length for each value.

Each vertex is of type Location. There is no limit on the number of vertices in a BoundarySeries although at least two are the minimum required. The vertices order is such that looking toward Earth center, they spiral in a clockwise direction away from the Earth[7]. Generally used for (planar) bounding boxes. However, it can support the specification of multifaceted (triangulated) volumes, as well; VTracker uses BoundarySeries in this manner to describe the bounding volume of a track. Figure 19 is an example of a shape and expected ordering of vertices. Vertices 1-4 are coplanar, as are vertices 5-8. The vertices encode in the following order: [1, 2, 3, 4, 5, 6, 7, 8].

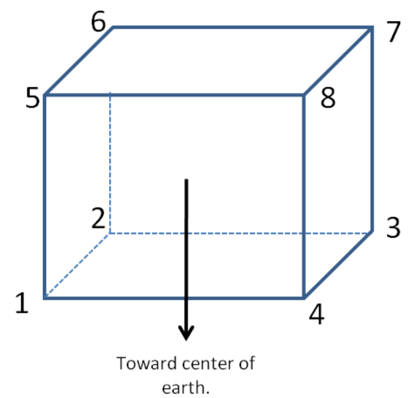

Figure 19: Example of a Simple Cubic Boundary

[7] The distances of the vertices from the Earth might not increase in a strictly monotonic fashion. One can imagine a sequence of vertices that exhibit “rollercoaster” behavior (without loops and crossovers).

Specifying two vertices imply opposite corners of a simple, planar bounding box, aligned with the Latitude-Longitude grid.

Requirement(s)	
ST 0903.4-93	A BoundarySeries shall contain at least two vertices.
ST 0903.4-94	Where a BoundarySeries contains only two vertices, the vertices shall be interpreted as opposite corners of a simple, planar bounding box, aligned with the Latitude-Longitude grid.

9 VTrack Local Set – Track Metadata

Whereas the VMTI LS describes detections of object movement associated within a single frame, the VTrack Local Set (LS) describes a set of detections that may span many frames presumed to be associated with a single, identified object. With few exceptions, the data items of the VTrack LS are the same as those defined for the VMTI LS. The VTrack LS adopts a "track-centric" view of the detection data, whereas the VMTI LS takes a "frame-centric" view. The data describing a detection are the same, regardless of the view taken, but their organization is different. Figure 20 presents a UML data model of the VTrack LS.

The VTrack LS provides another way to view track information. Examination and comparison of the two models should make the differences in organization readily apparent.

Nomenclature and notes regarding the VTrack model:

- Orange color denotes Local Sets; Blue indicate Pack structures; green is a mix of pack/LS
- The class name reflects the collective meaning of the attributes within the class. Class attributes are the informational items defining the class content. The attribute trait Name is a single word or phrase starting with a lowercase letter and in lower camel case. The attribute trait Type defines the data representation of the attribute value once the class has been instantiated
    - - <attribute> is for pack items e.g., -latitude in the Location Pack
    - - ##_<attribute> lists Local Set items e.g., -02_ontology in the Ontology Local Set
    - Multiplicities denote "Series" of Local Sets or Packs
    - Arrays ("[ ]") denote Series of simple types such as integers

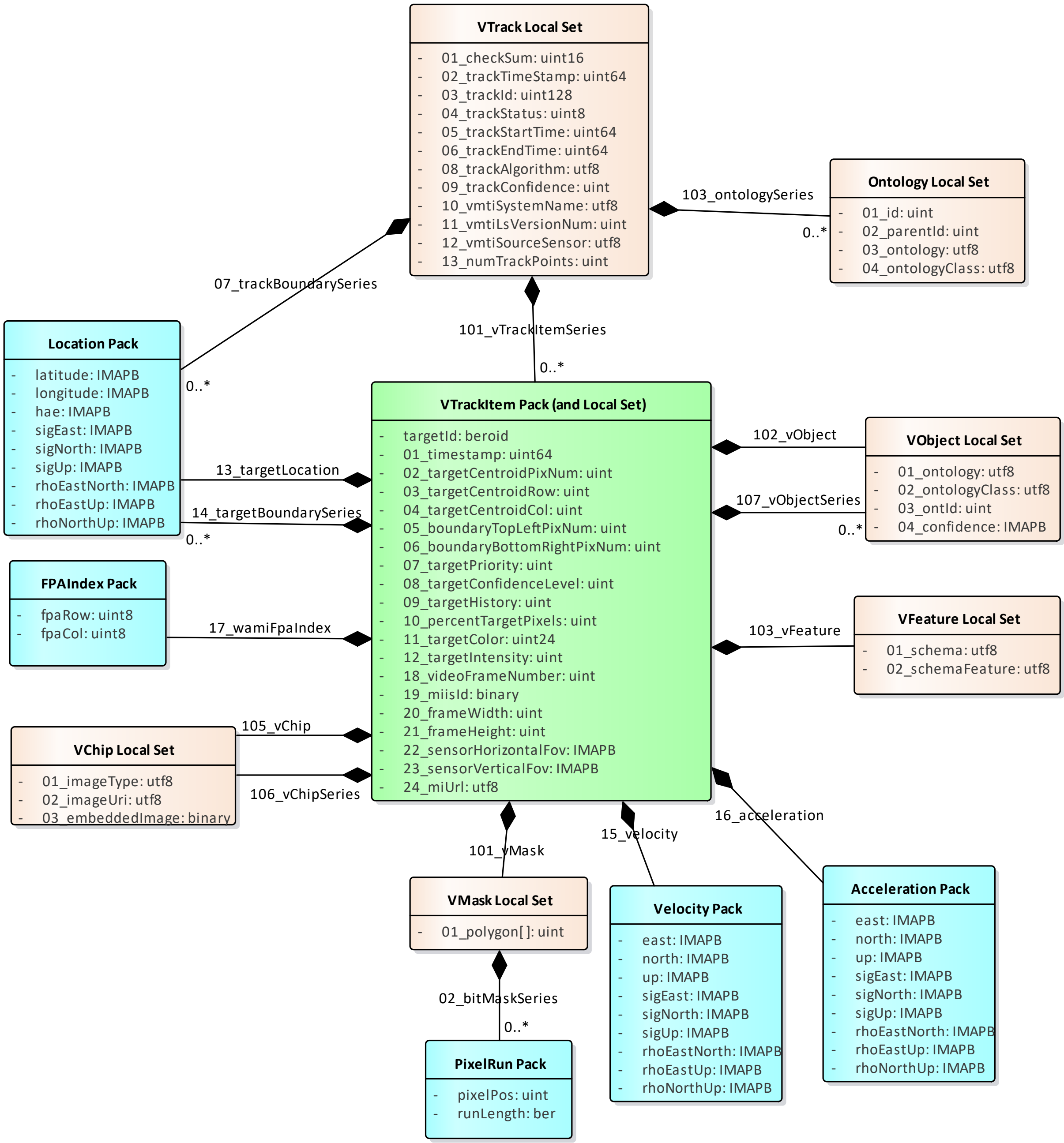

Figure 20: VTrack Local Set UML Model

Table 17 shows a high-level view of the VTrack LS.

Table 17. Structure of VTrack Local Set

VTrack Local Set			Description
TLV items			TLV triplets of data items
VTrackItemSeries			Series of VTrackItem Packs
VTrackItem Pack$_1$	• • •	VTrackItem Pack$_N$	Data for one track$_1$ to track$_N$
VMask LS	• • •	VMask LS	Delineate target perimeter
VObject LS	• • •	VObject LS	Class or Type of target
VFeature LS	• • •	VFeature LS	Features of target
VChip LS	• • •	VChip LS	Image pixels for the target
VChipSeries[8]	• • •	VChipSeries	Series of one or more VChip LS
OntologySeries			Series of ontologies used
Ontology LS			Ontology description
trackBoundarySeries			Boundary of track

The VMTI LS contains the structure VTargetSeries, which is a set of VTarget detections within an image frame. Similarly, the VTrack LS contains the VTrackItemSeries, structure, which is comprised of a set of VTrackItem detections associated with a track.

The VTrack LS and its constituent VTrackItem Packs provide a mechanism for reporting complete tracks, from the first detection of object movement to the latest. However, VTrack LS can also report track updates, including new detections that extend a previously reported track, or updates to previously reported track information.

Note that the VMTI LS contains VTracker[9], which is an alternative to VTrack LS to specify track metadata. VTracker contains spatial and temporal information for a given detection within a frame, including preceding detections of the same (presumed) object. However, VTrack LS maps more directly to NATO STANAG 4676, the NATO ISR Tracking Standard.[10]

The VTrack LS, shown in Table 18, consists primarily of metadata elements that appear in the VMTI LS (and its VTracker Pack), plus one additional data element, VTrackItemSeries.

VTrackItemSeries contains VTrackItem Packs. Each VTrackItem Pack describes a time-ordered sequence of points (e.g., VMTI detections) along the track of a moving object. Often, such track points connect using straight line segments to describe (an approximation of) the path taken by the moving object. However, this standard does not restrict interpretation of the points solely to this manifestation.

---

[8] Although permissible, it would be unusual to include both a VChip LS and a VChipSeries in a single VTarget Pack.

[9] Note the distinction between VTrack<u>er</u> and VTrack (no "er").

[10] STANAG 4676 Edition 1 currently lacks the extensive set of Motion Imagery feature information provided by VTrack to support object disambiguation and identification. Also, STANAG 4676 Edition 1 recognizes only XML as a data encoding method, which (because of its "verbosity") limits its use in bandwidth-limited communications environments. As STANAG 4676 matures to address these deficiencies, it may become the preferred representation.

Geographic coordinates in the VTrack LS are absolute, so the VTrack LS is always independent of MISB ST 0601.

The VTrack Local Set 16-Byte Universal Label "Key" registered in MISB ST 0807 [8] is:

06.0E.2B.34.02.03.01.01.0E.01.03.03.1E.00.00.00 (CRC 62593)

Table 18 summarizes the VTrack Local Set. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VTrack Local Set
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's KLV format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.
- Description: Items referenced to another table indicate tag ID names in those tables when different.

Table 18: VTrack Local Set

VTrack Local Set Key					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
1	checksum	See Table 2: VMTI Local Set			
2	trackTimeStamp	See Table 2: VMTI Local Set; This precisionTimeStamp indicates the time of the track report. By contrast, Track End Time (VTrack LS Tag 6) indicates the time of the most recent observation of the entity.			
3	trackId	See Table 9: VTracker Local Set			
4	trackStatus	See Table 9: VTracker Local Set detectionStatus			
5	trackStartTime	See Table 9: VTracker Local Set startTime			
6	trackEndTime	See Table 9: VTracker Local Set endTime			
7	trackBoundarySeries	See Table 9: VTracker Local Set BoundarySeries			
8	trackAlgorithm	See Table 9: VTracker Local Set algorithmId			
9	trackConfidence	See Table 9: VTracker Local Set confidenceLevel			
10	vmtiSystemName	See Table 2: VMTI Local Set			
11	vmtiLsVersionNum	See Table 2: VMTI Local Set			
12	vmtiSourceSensor	See Table 2: VMTI Local Set			
13	numTrackPoints	N/A	uint	V	The number of VTrackItem Packs contained in the VTrackItemSeries (Tag 101)
101	vTrackItemSeries	N/A	Series	V	Track metadata values, each of which is a VTrackItem Pack
103	ontologySeries	See Table 2: VMTI Local Set			

9.1 *VTrack LS:* Tag 101 VTrackItemSeries

A Series of one or more VTrackItem Packs.

9.2 VTrackItem Pack

The VTrackItem Pack contains data items from the VMTI LS and the VMTI VTarget Pack to describe not only the geographic location of a track point, but also a rich characterization of the moving object, including velocity, acceleration, appearance (features), and type. Furthermore, VTrackItem supports identification and reference ("linking") to the Motion Imagery essence in which the moving object appeared and detections of their movement.

The reference to – that is, the link to – the Motion Imagery essence is a Uniform Resource Identifier (URI), which uniquely identifies a Motion Imagery essence (i.e., a stream or file). The Motion Imagery essence URI may be a Uniform Resource Name (URN) such as a MIIS Core Identifier conformant to MISB ST 1204. Alternatively, the Motion Imagery essence URI may be a Uniform Resource Locator (URL) that not only uniquely identifies the Motion Imagery essence but also provides a means for locating the essence by describing its access mechanism (e.g., its network "location"). IETF RFC 3986 [15] defines URI syntax and a process for resolving URI references. Thus, the VTrackItem Pack contains two distinct metadata identifiers to reference the Motion Imagery essence: (1) Tag 19 Motion Imagery ID which is a URN conformant with MISB ST 1204 and (2) Tag 24 Motion Imagery URL which is a URL conformant with RFC 3986. If desired, both may appear.

Table 19 summarizes the VTrackItem Pack. The column designations are as follows:

- "Tag ID" indicates the Tag to use when specifying the item in the VTrackItem Pack
- "Name" is the label associated with the tag
- "Units" specifies the units of the data specified in the Value field
    - Units of "N/A" indicates no units apply
- "Format" indicates the item's KLV format for the Value
- "Len" indicates the nominal length of an item in bytes. See Section 7.1.2 for nomenclature of variable length assignments.
- Description: Items referenced to another table indicate tag ID names in those tables when different.

Table 19: VTrackItem Pack

VTrackItem Pack Key					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
N/A	targetId	See Table 4: VTarget Pack			
1	targetTimeStamp	See Table 2: VMTI Local Set precisionTiemStamp			
2	targetCentroidPixNum	See Table 4: VTarget Pack targetCentroid			
3	targetCentroidPixRow	See Table 4: VTarget Pack centroidPixRow			
4	targetCentroidPixCol	See Table 4: VTarget Pack centroidPixCol			
5	boundaryTopLeftPixNum	See Table 4: VTarget Pack boundaryTopLeft			
6	boundaryBottomRightPixNum	See Table 4: VTarget Pack boundaryBottomRight			
7	targetPriority	See Table 4: VTarget Pack			
8	targetConfidenceLevel	See Table 4: VTarget Pack			
9	targetHistory	See Table 4: VTarget Pack			
10	percentTargetPixels	See Table 4: VTarget Pack percentageOfTargetPIxels			

VTrackItem Pack Key					
Tag ID	**Name**	**Units**	**Format**	**Len**	**Description**
11	targetColor	See Table 4: VTarget Pack			
12	targetIntensity	See Table 4: VTarget Pack			
13	targetLocation	See Table 4: VTarget Pack			
14	targetBoundarySeries	See Table 4: VTarget Pack BoundarySeries			
15	velocity	See Table 9: VTracker Local Set			
16	acceleration	See Table 9: VTracker Local Set			
17	fpaIndex	See Table 4: VTarget Pack			
18	videoFrameNumber	See Table 2: VMTI Local Set motionImageryFrameNum			
19	miisId	See MIIS Core Identifier MISB ST 1204			
20	frameWidth	See Table 2: VMTI Local Set			
21	frameHeight	See Table 2: VMTI Local Set			
22	sensorHorizontalFov	See Table 2: VMTI Local Set vmtiHorizontalFov			
23	sensorVerticalFov	See Table 2: VMTI Local Set vmtiVerticalFov			
24	miUrl	N/A	utf8	V	A Uniform Resource Locator (URL) for the Motion Imagery essence, conformant with IETF RFC 3986
101	vMask LS	See Table 6: VMask Local Set			
102	vObject LS	See Table 7: VObject Local Set			
103	vFeature LS	See Table 8: VFeature Local Set			
105	vChip LS	See Table 11: VChip Local Set			
106	vChipSeries	See Table 4: VTarget Pack			
107	vObjectSeries	See Table 4: VTarget Pack			

Requirement(s)	
ST 0903.4-95	VTrackItem Motion Imagery ID (Tag 19) shall be a Motion Imagery Identification System Core Identifier that conforms to MISB ST 1204.
ST 0903.4-96	VTrackItem Motion Imagery URL (Tag 24) shall be a Uniform Resource Locator (URL) that conforms to IETF RFC 3986.

9.3 Required VTrack and VTrackItem Data

Both the VTrack LS and the VTrackItem Pack contain data that may not be required every time VTrack and its VTrackItem's are reported, either because the tracking system is incapable of producing such data or because the data changes infrequently. When sending metadata in a bandwidth-constrained environment, it is prudent to devote most of the available bandwidth to information that changes rapidly, sending less frequently changing information less often. For example, the location of a moving object should be provided at every opportunity, whereas an image chip (VChip) of that object might be provided "once in a while". With one exception, every time a track is reported or updated the data required is listed in Table 20.

Table 20: Required Metadata[11]

VTrack	VTrackItem	Tag ID	Name
X		01	checksum
X		02	trackTimeStamp (Precision Time Stamp)
X		03	trackId
	X	N/A	targetId
	X	01	targetTimeStamp
	X	13	targetLocation

The exception occurs for a VTrack report when Track Status is "Dropped." In this case, there might be no location information to report, so VTrackItem metadata may be omitted.

Requirement(s)	
ST 0903.5-110	The VTrack LS Checksum Tag 1 shall be present.
ST 0903.5-111	The VTrack LS Track Timestamp Tag 2 shall be present.
ST 0903.5-112	The VTrack LS Track ID Tag 3 shall be present.
ST 0903.5-113	Where VTrack Track Status Tag 4 is not "Dropped", VtrackItem Target ID Number shall be present.
ST 0903.5-114	Where VTrack Track Status Tag 4 is not "Dropped", VtrackItem Target Timestamp Tag 1 shall be present.
ST 0903.5-115	Where VTrack Track Status Tag 4 is not "Dropped", VtrackItem Target Location Tag 13 shall be present.

[11] These required data are consistent with those required by STANAG 4676.

10 Deprecated Requirements

ST 0903.4-02 (Deprecated)	All data encoded using the Key-Length-Value (KLV) encoding protocol shall comply with SMPTE ST 336. [defined in MISB ST 0107]
ST 0903.4-06 (Deprecated)	The Series type shall be a one-dimensional array of data elements, all of the same type, encoded as a Variable Length Pack. [defined in document already]
ST 0903.4-07 (Deprecated)	VTargetSeries shall be a Series of VTarget Packs. [defined in document]
ST 0903.4-39 (Deprecated)	Coordinates used for VMask LS Polygon (Tag 1) and VMask LS Bit Mask (Tag 2) shall be specified using pixel numbers calculated with the equation *Column* + ((*Row*-1) x frame width)), where numbering commences at 1 from the left for *Column* and from the top for *Row*, and where frame width is the number of columns in the image. [replaced with req's 100 & 101]
ST 0903.4-61 (Deprecated)	Defined-Length Truncation Packs shall be defined in accordance with MISB RP 0701. [defined in document]
ST 0903.4-64 (Deprecated)	The Location Structure shall be encoded as a Defined-Length Truncation Pack. [defined in document]
ST 0903.4-73 (Deprecated)	The Velocity Structure shall be encoded as a Defined-Length Truncation Pack. [defined in document]
ST 0903.4-74 (Deprecated)	The Velocity pack shall consist of a Velocity Components triplet, a Standard Deviations triplet, and a Correlation Coefficients triplet, in that order. [defined in document]
ST 0903.4-83 (Deprecated)	The Acceleration Structure shall be encoded as a Defined-Length Truncation Pack. [defined in document]
ST 0903.4-97 (Deprecated)	The VTrack LS and VTrackItem Pack metadata listed in MISB ST 0903.4 Table 20 shall be provided every time a track is reported or updated, unless the VTrack LS Track Status (Tag 4) indicates a status of "Dropped", in which case the VTrackItem metadata elements may be omitted. [replaced by req's 110-115]

11 Appendix A – Operational Considerations [Informative]

11.1 Bandwidth

Bandwidth management is of importance for VMTI (and tracking) systems, which produce a large amount of dynamic data at frame rate. In its simplest implementation, the VMTI LS may send thousands of targets to downstream systems by providing a Target ID Number and a Target Centroid Pixel Number within a Motion Imagery frame for each target. At the other end of the scale, the VMTI LS has scope to include multiple features about each target, image chips of the target, tracking information about the target, and numerous descriptive elements. The bandwidth overhead required to include all this information is very large – especially at 60 frames per second (FPS) or higher.

Motion Imagery sensors with frame rates up to 60 FPS are in use operationally, with faster systems expected in the future. In order to reduce bandwidth, data that changes infrequently need not update every frame but only often enough to ensure its available in any clip extracted from the Motion Imagery. Dynamic data should be delivered at a rate appropriate to the granularity of the intelligence provided. For example, for a target moving at 3 meters per second, a rate of 60 updates a second provides very little value over a rate of 20 updates per second. In that case, while the Motion Imagery frame rate might be 60 FPS, the VMTI update rate need only be 20 FPS.

The VMTI LS includes many elements that may not be available from onboard VMTI processors; thus, downstream processes will contribute "value add" elements to a basic VMTI stream. Calculations of bandwidth requirements should consider where in the VMTI workflow the data is added and the available bandwidth at that stage.

Bandwidth implications and bandwidth management is important in the design of systems generating VMTI data. Reducing metadata to a minimal effective configuration is good engineering. The VMTI LS leverages metadata available in other sets, such as MISB ST 0601. However, some metadata elements (VMTI Source Sensor, VMTI HFOV and VFOV) in the VMTI LS are comparable to those in ST 0601 LS (Image Source Sensor, HFOV, and VFOV). They are necessary, however, when VMTI operates on different Motion Imagery from that described by and/or included with the MISB ST 0601 data. Consider, for example, two bore-sighted sensors, where ST 0601 metadata describes the Motion Imagery from one ST 0601 sensors, but ST 0903 metadata describes VMTI detections from the Motion Imagery of a different sensor. (Note that ST 0601 metadata allows only one VMTI LS instance, precluding the carriage within it of detections from multiple Motion Imagery essences.)

Each VMTI process from a given sensor requires its own individual VMTI LS. That is, a VMTI LS should never contain a mixture of moving targets detected by different sensors. (The VChipSeries for a given detection may, however, contain image chips from multiple sensors.)

11.2 Co-located Bore-sighted Sensors

A system with multiple bore-sighted imagers within a single turret may send a Motion Imagery stream from one camera with a given field of view, synchronously with VMTI hits from other sensors with different fields of view. For example, a system containing a narrow field of view

(FOV) visible light sensor (EON), a wide FOV visible light sensor (EOW), and an infrared (IR) sensor may be transmitting a stream from the EON sensor and simultaneously include VMTI hits from both the EOW sensor and the IR sensor. This scenario necessitates a separate VMTI stream, consisting of distinct ST 0903 metadata for the IR and EOW cameras, carried along with the EON Motion Imagery. VMTI streams should be distinguishable by the VMTI Source Sensor field plus the VMTI Sensor Horizontal Field of View (HFOV) field.

Sophisticated VMTI systems may use the same Target ID Number to identify a common target detected by different sensors and retain the use of same Target ID Number temporally (that is, from one detection to another). Also, downstream processes (e.g., trackers and fusion systems) may reassign Target ID Numbers to identify a common target.

11.2.1 Independent Sensors

Each independent sensor system requires a separate MISB ST 0601 LS. The extra elements required in the VMTI LS to support multiple non-bore-sighted sensors would disproportionally increase bandwidth requirements for inclusion within a single ST 0601 LS packet. Given that ST 0601 does not support such cases anyway, the most appropriate solution is to generate individual VMTI LSs and ST 0601 streams for these sensors.

11.2.2 Large Volume Motion Imagery

Large Volume Motion Imagery (LVMI) systems present a problem not normally encountered with “traditional” airborne Motion Imagery sensors. LVMI systems cover large geographic extents and can detect thousands of simultaneous moving objects over several square kilometers of area. A standalone stream of VMTI or Track information (independent of Motion Imagery essence) could describe the detected moving objects and used to cue analysts as to which objects to monitor actively and which to leave to automated processes. This information can task “spotlights” which specify regions of interest to send accompanying Motion Imagery, which reduces data bandwidth. These spotlights in turn can carry VMTI or Track data for the moving objects within the scope of the spotlight imagery.

VMTI and Track data for the spotlight streams will be like the independent sensor’s paradigm in their packaging and processing. However, VMTI and Track data used for cueing takes a different approach. In this case there may be no associated Motion Imagery present. Cueing VMTI or Track is therefore self-contained, with no dependencies on other data streams.